






**ARCHITECTURE OF INPUT-OUTPUT PROCESSOR, WHICH CONNECTS  
INTER-BRIDGE OF PRIMARY COMPONENTS****Publication number:** RU2157000 (C2)**Publication date:** 2000-09-27**Inventor(s):** EHLLIOT GARBUS [US]; PITER SANGKHAGOVIT [US];  
MARK GOLDSCHMIDT [US]; NIK EHSKANDARI [US]**Applicant(s):** INTEL CORP [US]**Classification:****- international:** **G06F13/36; G06F3/02; G06F13/00; G06F13/14; G06F13/40;  
G06F15/00; G06F13/36; G06F3/02; G06F13/00; G06F13/14;  
G06F13/40; G06F15/00; (IPC1-7): G06F13/00; G06F3/02;  
G06F13/14; G06F15/00****- European:** G06F13/40D5**Application number:** RU19980100483 19960617**Priority number(s):** US19950490654 19950615**Also published as:** US5884027 (A) WO9700480 (A1) KR100263277 (B1) JP11513150 (T) EP0834135 (A1)

more &gt;&gt;

**Abstract of RU 2157000 (C2)**

computer engineering. SUBSTANCE: integral circuit for connection of first external inter-bridge bus for peripheral components to second external inter-bridge bus for peripheral components has two inner buses, bus bridge, local processor, local bus, memory controller. Processor system has main processor, peripheral device, two internal buses, bus bridge, local processor, local bus and memory controller. EFFECT: increased efficiency of input-output processor. 24 cl, 11 dwg, 46 tbl

Data supplied from the **esp@cenet** database — Worldwide



(19) **RU** <sup>(11)</sup> **2 157 000** <sup>(13)</sup> **C2**  
(51) МПК<sup>7</sup> **G 06 F 13/00, 3/02, 13/14, 15/00**

РОССИЙСКОЕ АГЕНТСТВО  
ПО ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

## (12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ РОССИЙСКОЙ ФЕДЕРАЦИИ

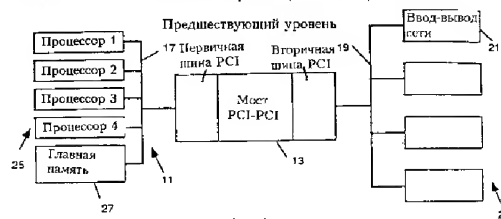
(21), (22) Заявка: 98100483/09, 17.06.1996  
(24) Дата начала действия патента: 17.06.1996  
(30) Приоритет: 15.06.1995 US 08/490,654  
(46) Дата публикации: 27.09.2000  
(56) Ссылки: US 5396602 A, 07.03.1995. SU 1644151 A1, 24.04.1991. SU 1683023 A1, 07.10.1991. US 5353417 A, 04.10.1994. US 5398244 A, 14.03.1995. US 4935894 A, 19.06.1990.  
(85) Дата перевода заявки РСТ на национальную фазу: 15.01.1998  
(86) Заявка РСТ: US 96/10451 (17.06.1996)  
(87) Публикация РСТ: WO 97/00480 (03.01.1997)  
(98) Адрес для переписки: 129010, Москва, ул. Большая Спасская 25, стр.3, ООО "Городисский и Партнеры", Емельянову Е.И.

(71) Заявитель:  
ИНТЕЛ КОРПОРЕЙШН (US)  
(72) Изобретатель: Эллиот ГАРБУС (US),  
Питер САНГХАГОВИТ (US), Марк ГОЛДШМИДТ (US), Ник ЭСКАНДАРИ (US)  
(73) Патентообладатель:  
ИНТЕЛ КОРПОРЕЙШН (US)

### (54) АРХИТЕКТУРА ПРОЦЕССОРА ВВОДА-ВЫВОДА, КОТОРЫЙ ОБЪЕДИНЯЕТ МОСТ МЕЖСОЕДИНЕНИЯ ПЕРВИЧНЫХ КОМПОНЕНТ

(57) Изобретение относится к области архитектуры компьютерной системы. Техническим результатом является повышение производительности процессора ввода/вывода. Данный результат достигается тем, что интегральная схема соединения первой внешней шины межсоединения периферийных компонент со второй внешней шиной межсоединения периферийных компонент содержит две внутренние шины, шинный мост, локальный процессор, локальную шину, контроллер памяти. Процессорная система содержит главный

процессор, периферийное устройство, две внутренние шины, шинный мост, локальный процессор, локальную шину, контроллер памяти. 2 с. и 22 з.п.ф-лы, 11 ил., 46 табл.



Фиг. 1



RUSSIAN AGENCY  
FOR PATENTS AND TRADEMARKS

(19) **RU** <sup>(11)</sup> **2 157 000** <sup>(13)</sup> **C2**  
(51) Int. Cl.<sup>7</sup> **G 06 F 13/00, 3/02, 13/14, 15/00**

(12) **ABSTRACT OF INVENTION**

(21), (22) Application: 98100483/09, 17.06.1996  
(24) Effective date for property rights: 17.06.1996  
(30) Priority: 15.06.1995 US 08/490,654  
(46) Date of publication: 27.09.2000  
(85) Commencement of national phase: 15.01.1998  
(86) PCT application:  
US 96/10451 (17.06.1996)  
(87) PCT publication:  
WO 97/00480 (03.01.1997)  
(98) Mail address:  
129010, Moskva, ul. Bol'shaja Spasskaja 25,  
str.3, OOO "Gorodisskij i Partnery",  
Emel'janovu E.I.

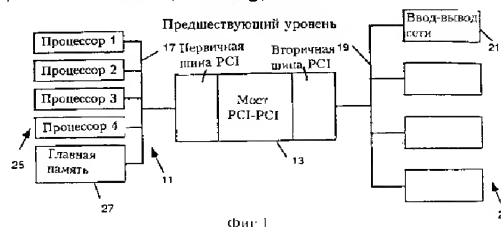
(71) Applicant:  
INTEL KORPOREJShN (US)  
(72) Inventor: Ehlliott GARBUS (US),  
Piter SANGKhAGOVIT (US), Mark GOLDSShMIDT  
(US), Nik EhSKANDARI (US)  
(73) Proprietor:  
INTEL KORPOREJShN (US)

(54) **ARCHITECTURE OF INPUT-OUTPUT PROCESSOR, WHICH CONNECTS INTER-BRIDGE OF PRIMARY COMPONENTS**

(57) Abstract:

FIELD: computer engineering. SUBSTANCE: integral circuit for connection of first external inter-bridge bus for peripheral components to second external inter-bridge bus for peripheral components has two inner buses, bus bridge, local processor, local bus, memory controller. Processor system has main processor, peripheral device, two internal buses, bus bridge, local processor, local bus and memory controller. EFFECT:

increased efficiency of input-output processor. 24 cl, 11 dwg, 46 tbl



Изобретение относится к области архитектуры компьютерной системы. Более конкретно, данное изобретение относится к интеллектуальному шинному мосту для внедрения интеллектуальной подсистемы ввода/вывода в компьютерные и серверные системы.

Высокопроизводительные компьютерные системы обычно включают в себя отдельные подсистемы ввода/вывода. Такая подсистема ввода/вывода обычно включает в себя микропроцессор, который выполняет функции ввода/вывода и который отделен от устройства, которое может быть названо главным или основным микропроцессором. Например, такая подсистема ввода/вывода может выполнять сложные интерфейсные функции при обмене в сети или функции управления диском для компьютерной системы.

Обычно подсистема ввода/вывода включает набор специализированных устройств ввода/вывода, соединенных для обмена на шине компонентов. Процессор в такой подсистеме ввода/вывода обычно выполняет функции ввода/вывода по шине без помехи для операций, выполняемых другими процессорами в компьютерной системе. Такая изоляция транзакций ввода/вывода на шине обычно дает возможность увеличить производительность основного процессора или процессоров в такой компьютерной системе. Такая архитектура является общей в универсальных компьютерных системах, в которых процессор и подсистема ввода/вывода называются каналом ввода/вывода.

С появлением компьютерных систем, использующих микропроцессоры, особенно систем "сервер/клиент", увеличивается потребность в более мощных микропроцессорах для обеспечения возможности построения более мощных систем "сервер/клиент". Это требование частично удовлетворяется с помощью объединения множества микропроцессоров в единую систему 11, которая изображена на фиг. 1. Другая проблема, которая возникает при необходимости объединить большее количество устройств ввода/вывода для реализации более мощных систем "сервер/клиент", заключается в том, что стандартные компонентные шины, которые соединяют подсистемы ввода/вывода с другими элементами компьютерной системы, обычно налагают ограничения по электрической нагрузке. Такие ограничения по электрической нагрузке устанавливают пределы по количеству компонентов, подключенных к стандартной шине компонентов. Например, одна известная стандартная шина требует, чтобы каждое соединительное устройство на системной соединительной шине компонентов представляло собой только одну электрическую нагрузку. Такие ограничения по электрической нагрузке гарантируют, что качество сигнала на полностью нагруженной шине является достаточным для надежной работы.

В этой связи, так как некоторые подсистемы ввода/вывода требуют большого количества обмениваемых через локальную шину компонентов, которое может превышать требования к электрической

нагрузке, налагаемые на каждый соединитель стандартной шины компонентов, то подсистема ввода/вывода может также содержать схему 13 шинного моста (развязывающего устройства шин), которая подсоединяет локальную шину 17 компонентов к другим шинам компонентов 19 в компьютерной системе, которая подсоединена к сети, такой как локальная сеть (LAN), посредством сетевой платы 21 ввода/вывода, или к запоминающим устройствам посредством контроллеров 23 SCSI.

#### Приложение 2

(интерфейс малых вычислительных машин). Такой шинный мост электрически изолирует микропроцессор или микропроцессоры 25, память 27 и компоненты подсистемы ввода/вывода от других компонентных шин. Такая схема шинного моста дает возможность подсистеме ввода/вывода содержать большое количество компонентов, требуемых для реализации функций ввода/вывода, в то же время удовлетворяя требованиям к электрической нагрузке на других компонентных шинах.

Кроме того, микропроцессор или процессоры в системе 11 обычно должны конфликтовать с другими исполнительными устройствами шины, подсоединенными к компонентной шине. Такие конфликты на шине обычно снижают производительность микропроцессора или микропроцессоров, в то же время выполняя функции ввода/вывода для подсистемы ввода/вывода.

Первым объектом настоящего изобретения является

Интегральная схема (31) соединения первой внешней шины межсоединения периферийных компонент (PCI) со второй внешней шиной межсоединения периферийных компонент (PCI), содержащая первую внутреннюю шину для подсоединения к указанной первой внешней PCI шине,

вторую внутреннюю шину для подсоединения к указанной второй внешней PCI шине,

шинный мост PCI-PCI (32), имеющий первый шинный интерфейс (71), подсоединенный к указанной первой внутренней шине, и второй шинный интерфейс, подсоединенный к указанной второй внутренней шине, причем указанный шинный мост позволяет пропускать транзакции между первой и второй внешними PCI шинами,

локальный процессор (34), имеющий пространство для локального адреса, подсоединенный к указанному шинному мосту (32) для обработки транзакций, принятых от внешних источников, и транзакций, поступающих на указанный первый шинный интерфейс и указанный второй шинный интерфейс,

локальную шину (41) для прохождения данных между указанным процессором (34) и памятью (33), доступ к которой осуществляется в указанном пространстве локального адреса,

контроллер (47) памяти для управления записями и считываниями из указанной памяти с использованием данных и адресов на указанной локальной шине.

Интегральная схема по настоящему



изобретению может дополнительно содержать

первую схему (43a) преобразования адреса, подсоединенную к указанной локальной шине (41) и указанной первой внутренней шине, приспособленную для преобразования адресов входящих транзакций на указанной первой внутренней шине в адреса, используемые указанным локальным процессором (34), и для преобразования адресов исходящих транзакций на указанной локальной шине (41) в адреса, используемые устройствами, подсоединенными к указанной первой внешней шине,

вторую схему (43b) преобразования адреса, подсоединенную к указанной локальной шине и указанной второй внутренней шине, приспособленную для преобразования адресов входящих транзакций на указанную вторую внутреннюю шину, в адреса, используемые указанным средством локального процессора, и для преобразования адресов исходящих транзакций на указанной локальной шине в адреса, используемые устройствами, подключенными к указанной второй внешней шине.

Кроме этого, интегральная схема (31) по настоящему изобретению может дополнительно содержать

схему (45) передачи сообщений, подсоединенную к указанной первой схеме (43a) преобразования адреса, приспособленную для формирования прерывания для использования указанным локальным процессором (34), когда новые данные поступают на указанную первую внутреннюю шину, и для формирования прерывания, по меньшей мере, на одной линии прерывания указанной первой внутренней шины, когда указанный локальный процессор (34) помещает данные на указанную локальную шину (41) для использования устройством, подсоединенным к указанной внешней шине,

первый контроллер (51a) прямого доступа к памяти (DMA), подсоединенный к указанной локальной шине (41) и указанной первой внутренней шине для передачи блоков данных между указанной первой внутренней шиной и указанной локальной памятью (33),

второй контроллер (51b) прямого доступа к памяти (DMA), подсоединенный к указанной локальной шине (41) и указанной второй внутренней шине для передачи блоков данных между указанной второй внутренней шиной и указанной локальной памятью (33),

первое средство (55a) арбитража шины для управления доступом к указанной первой внутренней шине между указанным первым шинным интерфейсом и указанной первой схемой (43a) преобразования адреса,

второе средство (55b) арбитража шины для управления доступом к указанной второй внутренней шине между указанным вторым шинным интерфейсом и указанной второй схемой (43b) преобразования адреса,

средство (55a) арбитража локальной шины для управления доступом к указанной локальной шине (41) от указанного локального процессора (34), указанной первой схемы (43a) преобразования адреса и указанной второй схемы (43b) преобразования адреса, контроллер (67) прерываний

периферийных устройств для маршрутизации множества источников прерываний на вход единственного прерывания к указанному локальному процессору (34),

регистр (123, 125) состояния, определяющий, какой из указанного множества источников прерывания вызвал указанное единственное прерывание, шину межсоединения интегральных схем ( $I^2C$ ) для подсоединения к внешним датчикам, подходящим для управления системы с использованием указанной интегральной схемы,

интерфейс (61) межсоединения интегральных схем ( $I^2C$ ) для разрешения указанному локальному процессору (34) служить в качестве ведущего и подчиненного устройства, постоянно находящегося на указанной шине  $I^2C$ ,

шину усовершенствованного программируемого контроллера прерываний (APIC) для подключения к внешним процессорам,

интерфейс (63) шины APIC для разрешения обмена между указанными внешними процессорами и указанным локальным процессором (34).

Вторым объектом настоящего изобретения является процессорная система, включающая в себя, по меньшей мере, один главный процессор (25), подсоединенный к первичной шине (17) межсоединения периферийных компонент (PCI) и, по меньшей мере, одно периферийное устройство (21, 23), подсоединенное к вторичной PCI шине (19), содержащая в единственной интегральной схеме

первую внутреннюю шину, подсоединенную к указанной первой PCI шине,

вторую внутреннюю шину, подсоединенную к указанной второй PCI шине,

шинный мост (32), имеющий первый шинный интерфейс, подсоединенный к указанной первой внутренней шине, и второй шинный интерфейс, подсоединенный к указанной второй внутренней шине, причем указанный шинный мост позволяет пропускать транзакции между указанной первичной PCI шиной и указанной вторичной PCI шиной,

локальный процессор (34), подсоединенный к указанному шинному мосту для обработки транзакций, принятых от внешних источников, и транзакций, поступающих на указанный первый шинный интерфейс и указанный второй шинный интерфейс,

локальную шину (41) для прохождения данных между указанным процессором и памятью,

контроллер (47) памяти для управления записями и считываниями из указанной внешней памяти с использованием данных и адресов на указанной локальной шине.

В процессорной системе согласно настоящему изобретению указанная интегральная схема (31) может дополнительно содержать

схему (45) передачи сообщений, подсоединенную к указанной первой схеме (43a) преобразования адреса, приспособленную для формирования прерывания для использования указанным локальным процессором (34), когда новые

данные поступают на указанную первую внутреннюю шину, и для формирования прерывания, по меньшей мере, на одной линии прерывания указанной первой внутренней шины, когда указанный локальный процессор (34) помещает данные на указанную локальную шину (41) для использования устройством, подсоединенным к указанной первой PCI шине,

первый контроллер (51a) прямого доступа к памяти (DMA), подсоединенный к указанной локальной шине и указанной первой внутренней шине для передачи блоков данных между указанной первой внутренней шиной и указанной локальной памятью,

второй контроллер DMA (51b), подсоединенный к указанной локальной шине и указанной второй внутренней шине для передачи блоков данных между указанной второй внутренней шиной и указанной локальной памятью,

первое средство (55a) арбитража шины для управления доступом к указанной первой внутренней шине между указанным первым интерфейсом шины и указанной первой схемой (43a) преобразования адреса,

второе средство (55b) арбитража шины для управления доступом к указанной второй внутренней шине между указанным вторым интерфейсом шины и указанной второй схемой (43b) преобразования адреса,

средство (55a) арбитража локальной шины для управления доступом к указанной локальной шине (41) от указанного локального процессора (34), указанной первой схемы (43a) преобразования адреса и указанной второй схемы (43b) преобразования адреса,

контроллер (67) прерываний периферийных устройств для маршрутизации множества источников прерываний на вход единственного прерывания к указанному локальному процессору (34),

регистр (123, 125) состояния, определяющий, какой из указанного множества источников прерывания вызвал указанное единственное прерывание,

шину межсоединения интегральных схем ( $I^2C$ ) для подключения к внешним датчикам, подходящим для управления системы с использованием указанной интегральной схемы,

( $I^2C$ ) интерфейс (61) для разрешения указанному локальному процессору (34) служить в качестве ведущего и подчиненного устройства, находящегося на указанной шине  $I^2C$ ,

шину усовершенствованного программируемого контроллера прерываний (APIC) для подключения к внешним процессорам,

интерфейс (63) APIC для разрешения обмена между указанными внешними процессорами и указанным локальным процессором (34),

первое средство (55a) арбитража шины для управления доступами к указанной первой внутренней шине от указанного первого интерфейса шины, указанной первой схемы (43a) преобразования адреса и указанного первого контроллера (51a) прямого доступа к памяти (DMA),

второе средство (55b) арбитража шины для управления доступом к указанной второй внутренней шине от указанного второго интерфейса шины, указанной второй схемы

(43b) преобразования адреса и указанного второго контроллера DMA (51b),

первое средство (55a) арбитража шины для управления доступами к указанной первой внутренней шине от указанного первого интерфейса шины, указанной первой схемы (43a) преобразования адреса и указанного первого средства (51a) контроллера DMA,

второе средство (55b) арбитража шины для управления доступами к указанной второй внутренней шине от указанного второго интерфейса шины, указанной второй схемы (43b) преобразования адреса и указанного второго контроллера DMA (51b).

Настоящее изобретение является многофункциональным устройством, которое внедряет высокопроизводительный процессор в шинный мост PCI-PCI (межсоединения периферийных компонентов) (P2P). На фиг.2 показано, как изобретение объединяет высокопроизводительный процессор, такой как процессор 80960 JF, производимый заявителем-фирмой Intel Corporation (локальный процессор), шинный мост PCI-PCI 32, устройство преобразования адреса "шина PCI - процессор", контроллер прямого доступа к памяти (DMA), контроллер памяти, устройство арбитража вторичной PCI шины, интерфейсное устройство шины для межсоединений интегральных схем ( $I^2C$ ), интерфейсное устройство шины усовершенствованных программируемых прерываний (APIC) и устройство передачи сообщений в единую систему 31, которое использует локальную память 33. Система является интегрированным процессором, который адресует запросы интеллектуальных прикладных задач ввода/вывода и помогает снизить стоимость интеллектуальной системы ввода/вывода.

Шина PCI (межсоединения периферийных компонент) является промышленным стандартом (PCI Local Bus Specification, "Спецификация локальной шины PCI", версия 2.1), высокопроизводительной системной шиной с малым временем задержки. Мост PCI-PCI предоставляет канал соединения между двумя независимыми 32-разрядными шинами PCI и предоставляет возможность преодоления ограничений на электрическую нагрузку шины PCI. Добавление локального процессора вносит свойства интеллектуальности шинному мосту PCI. Локальный процессор и другие функциональные блоки, находящиеся в изображенном на фиг. 3 пунктирной линией прямоугольнике, образуют блок-схему устройства, которое далее будет называться P2P процессором.

P2P процессор является многофункциональным PCI-устройством. Функция 0 соответствует мостовому устройству PCI-PCI. Функция 1 соответствует устройству преобразования адреса. P2P процессор имеет область (пространство) конфигурации PCI, доступную через первичную шину PCI.

В предпочтительном варианте осуществления локальный процессор 34 является процессором 80960 JF, который является элементом семейства микропроцессоров Intel i960. Процессор 80960 JF является встроенным в P2P процессор без функциональной модификации. "Руководство

пользователя по микропроцессорам i960" от Intel Corporation представляет дополнительные подробности, хотя вся необходимая для практического использования изобретения информация, касающаяся локального процессора, приводится в данном описании.

Локальный процессор работает вне своего собственного 32-битного адресного пространства и не в адресном пространстве PCI. Память на шине локального процессора может быть

- сделана видимой для адресного пространства PCI,
- сохранена принадлежащей локальному процессору,
- объединенной указанных двух.

Шина 41 локального процессора

Шина локального процессора подсоединяет линии ввода/вывода к процессору P2P для обеспечения доступа шины к внешним устройствам. Процессор P2P обеспечивает поддержку арбитража локальной шины.

Устройства 43a и 43b преобразования адреса и устройство 45 передачи сообщений

Устройство преобразования адреса разрешает PCI-транзакциям прямой доступ к локальной памяти 33 локального процессора. Локальный процессор 34 имеет прямой доступ к обеим PCI шинам. Преобразование адреса осуществляется для транзакций между адресным пространством PCI и адресным пространством локального процессора. Преобразование адреса управляется с помощью программируемых регистров, доступных со стороны PCI интерфейса и со стороны локального процессора, что обеспечивает гибкость в отображении двух адресных пространств. Устройство 45 передачи сообщений предоставляет механизм передачи данных между системой PCI и локальным процессором и оповещения соответствующей системы о поступлении новых данных посредством прерывания. Устройство обмена сообщениями может быть использовано длясылки и приема сообщений.

Мостовое устройство 32 PCI- PCI

Мостовое устройство PCI-PCI соединяет две независимые шины PCI. Мост позволяет некоторым шинным транзакциям в одной PCI шине поступать в другую PCI шину. Он также допускает полностью независимую работу PCI шины, включая независимую синхронизацию. Специализированные очереди данных поддерживают высокую пропускную способность PCI шин. Поддерживается 64-разрядная адресация посредством двойного адресного цикла PCI (DAC).

Мост PCI-PCI имеет выделенное пространство PCI конфигурации, которое доступно от первичной PCI шины.

Мост PCI-PCI в процессоре P2P является полностью совместимым со спецификацией архитектуры моста PCI-PCI (PCI to PCI Bridge Architecture Specification), версия 1.0, опубликованной PCI Special Interest Group.

Приватные PCI-устройства

P2P процессор по своей конструкции открыто поддерживает приватные PCI-устройства, которые могут использовать вторичную PCI шину, однако избегая определения программным обеспечением PCI-конфигурации. Мост 32 PCI-PCI и

вторичное устройство 43b преобразования адреса работают совместно для сокрытия приватных устройств от циклов PCI-конфигурации и для разрешения таким устройствам использовать приватное адресное пространство PCI. Эти устройства могут быть сконфигурированы вторичным устройством преобразования адреса с помощью обычных циклов PCI-конфигурации.

Встроенный контроллер памяти 47

Встроенный контроллер памяти обеспечивает прямое управление внешних систем памяти. Обеспечивается поддержка для DRAM (динамической памяти со случайным доступом), SRAM (статической памяти со случайным доступом), ROM (памяти только для чтения) и памяти на ЭСППЗУ (электрически стираемого программируемого постоянного запоминающего устройства) с параллельным стиранием. Встроенный контроллер памяти обеспечивает непосредственный соединительный интерфейс с памятью 33, обычно не требующей внешних логических схем. Его отличает программируемый выбор микросхемы, генератор режима ожидания и байт четности.

Внешняя память может быть сконфигурирована как PCI-адресуемая память или как приватная память локального процессора.

Контроллер 51a и 51b прямого доступа к памяти

Контроллер прямого доступа к памяти (DMA) разрешает высокоскоростную передачу данных с малыми задержками между исполнительными устройствами PCI шины и локальной памятью.

Существует три отдельных канала DMA для регулирования передачи данных. Два канала выделены для передачи данных первичной PCI-шины, и один канал выделен для передачи данных вторичной PCI- шины. Контроллер DMA поддерживает передачу сцепленных и невыровненных данных. Он является программируемым только через локальный процессор 34.

Устройство 53 арбитража вторичной PCI-шины

Устройство арбитража вторичной PCI-шины обеспечивает PCI-арбитраж (разрешение конфликтов) для вторичной PCI-шины. Реализован алгоритм равнодоступности с программируемыми приоритетами. Имеются шесть пар PCI-сигналов запросов и предоставлений. Устройство арбитража может быть заблокировано для разрешения внешнего арбитража.

Устройства 55a, 55b и 55с арбитража внутренней PCI-шины и локальной шины

Процессор P2P содержит два устройства внутреннего арбитража, которые управляют доступом к внутренним PCI шинам в устройстве, а именно, первичное внутреннее PCI-устройство 55a арбитража, которое разрешает конфликты для первичного интерфейса моста, первичного ATU, DMA канала 0 и DMA канала 1. Вторичное внутреннее PCI-устройство 55b арбитража разрешает конфликты для вторичного интерфейса моста, вторичного ATU и DMA канала 2. Каждое PCI-устройство внутреннего арбитража использует схему арбитража с фиксированным циклом, в которой каждое

устройство на шине имеет равный приоритет.

P2P процессор также требует механизма арбитража для монопольного управления локальной шиной. Устройство 57 арбитража локальной шины (LBAU) реализует алгоритм равнодоступности, который допускает каждому "хозяину" шины возможность захвата управления локальной шиной. Алгоритм объединяет циклическую схему опроса и механизм с приоритетами.

Интерфейсное устройство шины I<sup>2</sup>C 61

Интерфейсное устройство шины I<sup>2</sup>C для межсоединений интегральных схем позволяет локальному процессору служить в качестве ведущего и подчиненного устройства, постоянно находящегося на шине I<sup>2</sup>C. Шина I<sup>2</sup>C является последовательной шиной, разработанной Philips Corporation, и состоит из двухпроводного интерфейса. Шина позволяет P2P процессору обмениваться с другими I<sup>2</sup>C периферийными устройствами и микроконтроллерами для осуществления функций управления системой. Это требует минимального аппаратного обеспечения для экономичной системы для передачи информации о состоянии и надежности на подсистему ввода/вывода к внешнему устройству.

Интерфейсное устройство 63 шины APIC

Интерфейсное устройство шины APIC обеспечивает интерфейс с трехпроводной шиной усовершенствованного программируемого контроллера прерываний (APIC), которая позволяет эмулировать APIC ввода/вывода в программном обеспечении. Сообщения о прерываниях могут быть посланы на шину и могут быть приняты сообщениями EO1 (конец прерывания).

Маршрутизация прерываний (67)

Имеются четыре входа прерываний PCI, которые могут быть направлены или ко входам прерывания локального процессора или выходным линиям прерывания PCI.

ТЕРМИНОЛОГИЯ И СОГЛАШЕНИЯ

Цифровые обозначения

Все числа, представленные ниже, приводятся в системе счисления с основанием 10, если не указано обратное. В тексте число по основанию 16 представляется как "nnnH", где "H" обозначает 16-теричную систему счисления. Двоичные числа изображаются с индексом 2.

Поля

Защищенная область в структуре данных является такой, которую процессор не может использовать. Защищенные области могут быть использованы программным обеспечением, процессор не будет модифицировать такие области.

Защищенная область является областью, которая может быть использована реализацией. Если начальное значение защищенной области задается программным обеспечением, это значение должно быть равно нулю. Программное обеспечение не может модифицировать защищенные области или зависеть от любых значений в защищенных областях.

Область только для чтения может быть считана для возвращения текущего значения. Записи в области "только для чтения" пресекаются как недопустимые операции и не изменят ни текущее значение, ни результат в условии ошибки.

Область считывания/очистки также может

быть считана для возвращения текущего значения. Запись данных, имеющих значение 0, в область считывания/очистки не вызовет никаких изменений в области. Запись данных, имеющих значение 1, в область считывания/очистки вызовет очистку области (установку значения данных 0). Например, если область считывания/очистки имеет значение F0H, а записывают данные 55H, получившееся значение будет равно A0H.

Терминология

Дополнительно при описании архитектуры P2P используется следующая терминология:

Вниз - на PCI шине или по направлению к PCI шине с более высоким номером (после конфигурации)

DWORD - слово данных размерностью 32 бита

Главный процессор - процессор, расположенный вверх от процессора P2P

Локальная шина - шина локального процессора

Локальная память - подсистема памяти на локальной шине

Вверх - на PCI шине или по направлению к PCI шине с меньшим номером (после конфигурации)

Краткое описание чертежей

Фиг.1 - диаграмма, изображающая систему, использующую известный мост PCI - PCI,

фиг. 2 - диаграмма, изображающая систему, использующую новый мост PCI - PCI с процессором, согласно настоящему изобретению,

фиг.3 - диаграмма, изображающая изобретенный P2P процессор,

фиг.4 изображает направления, в которых передаются транзакции между первичным и вторичным адресными пространствами моста PCI - PCI,

фиг.5 - блок-схема моста PCI-PCI,

фиг. 6 - блок-схема первичного и вторичного устройств ATU,

фиг. 7 - диаграмма, изображающая соединения DMA контроллеров с различными шинами,

фиг. 8 - блок-схема контроллера памяти,

фиг. 9 - диаграмма, изображающая соединения между локальным процессором и PPIC,

фиг. 10 - диаграмма, изображающая соединения устройств внутреннего арбитража PCI,

фиг. 11 - блок-схема интерфейсного устройства шины I<sup>2</sup>C и его интерфейс с локальной шиной.

ПОДРОБНОЕ ОПИСАНИЕ ИЗОБРЕТЕНИЯ

Изобретение описывается ниже в терминах его функциональных блоков, приведенных на фиг.3.

Локальный процессор

Ниже приводится описание микропроцессора 80960 JF, используемого в качестве локального процессора в процессоре P2P. Описывается, как конфигурируют процессор 80960 JF или, с другой стороны, отличие от описания части в "Руководстве пользователя микропроцессора i960 Jx".

Обзор

Процессор 80960 JF реализуется без функциональных изменений в процессоре P2P, т.е. никакая внутренняя логика не изменяется. Обратитесь к "Руководству

пользователя микропроцессора i960 Jx" для получения более подробной информации о процессоре 80960 JF.

#### Особенности

Основными особенностями процессора 80960 JF являются следующие:

- высокопроизводительное ядро исполнения инструкций
- устанавливаемая двумя способами ассоциативная кэш-память инструкций объемом 4 Кбайт,
- кэш-память данных прямого отображения объемом 2 Кбайт,
- 32 целочисленных 32-битовых регистра,
- программируемый контроллер шины,
- 1 Кбайт внутренней памяти типа RAM для данных,
- локальная регистровая кэш-память, предоставляющая память до 8 наборов локальных регистров,
- усовершенствованный контроллер прерываний,
- два 32-битовых таймера.

#### ОТЛИЧИЯ

Далее следует описание конструктивных решений системы, которые делают процессор 80960J таким, как он используется в процессоре P2P.

#### Область памяти

Так как регистры отображения периферийной памяти процессора P2P являются 32-битовыми, то область памяти 0 и 1 должна обозначать 32-битовую область. Поэтому, регистр PMCON0\_1 должен иметь биты разрядности шины, установленные в 102, указывая на 32-битовую разрядность шины.

#### Шина

Для получения оптимальной производительности от DMA-обращений "хозяинам" шины, отличным от локального процессора, позволено иметь неограниченные длины пакетов на шине локального процессора. Адрес, однако, не будет увеличиваться для пакетов длиннее 4 слов. Это предполагает, что контроллеры памяти на локальной шине должны увеличивать адрес для каждого обращения в пакете.

#### МОСТОВОЕ УСТРОЙСТВО PCI-PCI

##### Обзор

Мостовое устройство 32 PCI-PCI является устройством, которое допускает расширение шины PCI свыше ее физических ограничений из 10 электрических PCI-нагрузок. Мостовое устройство использует концепцию иерархических шин, в которой каждая шина в иерархии является электрически отдельной, но в то же время все шины в иерархии логически являются одной шиной. Мостовое устройство PCI- PCI не увеличивает пропускную способность шины PCI, оно только позволяет расширить шину для прикладных задач, требующих большего количества компонентов ввода/вывода, чем допускает электрическая спецификация PCI.

Мостовое устройство PCI-PCI предоставляет

независимые 32-битовые первичную и вторичную PCI шины с поддержкой одновременных операций в любом направлении,

раздельные адресные пространства памяти ввода/вывода со вторичной стороны моста,

два 64-байтовых буфера пересылки и для направляемых вверх и для направляемых вниз транзакций,

палитру цветов VGA и VGA-совместимую адресацию на вторичной шине, режим 64-битовой адресации от вторичного PCI-интерфейса, конфигурацию приватного устройства и адресное пространство для приватных PCI устройств на вторичной PCI-шине, специальный режим работы, который разрешает положительное (прямое) декодирование на первичном и вторичном интерфейсах.

#### Теория работы

Мостовое устройство работает как устройство фильтрации адресов между первичной и вторичной PCI шинами. PCI поддерживает три раздельных адресных пространства:

- адресное пространство памяти размером 4 Гбайт,
- адресное пространство ввода/вывода размером 64 Кбайт (с 16-битовой адресацией),
- отдельное пространство конфигурации.

Мост PCI-PCI является

запрограммированным с непрерывным диапазоном адресов в адресных пространствах памяти и ввода/вывода, которые затем становятся адресным пространством вторичной PCI. Любой адрес, имеющийся с первичной стороны моста, который попадает в запрограммированное вторичное пространство, передается от первичной стороны во вторичную, в то же время адреса вне вторичного пространства мостом игнорируются. Вторичная сторона моста работает наоборот по отношению к первичной стороне, игнорируя любые адреса в запрограммированном вторичном адресном пространстве и передавая любые адреса вне вторичного пространства первичной стороне, как изображено на фиг. 4.

Первичный и вторичный интерфейсы PCI-моста каждый реализует ведущее устройство и устройство-адресат в соответствии с PCI спецификацией 2.1. PCI-транзакция, инициированная на одной стороне моста, будет адресовать инициирующий интерфейс моста шины в качестве адресата, и транзакция должна быть завершена шинным интерфейсом адресата, работающим как ведущее устройство. Мост является прозрачным для PCI-устройств с любой стороны.

Мостовое устройство PCI-PCI процессора P2P придерживается, как минимум, требуемых особенностей, указанных в "Спецификации архитектуры моста PCI-PCI", версия 1.0 и "Спецификации локальной шины PCI", версия 2.1. Далее следует описание функциональных особенностей моста и, где необходимо, будет осуществляться ссылка на спецификации моста PCI-PCI и шины PCI.

#### Описание архитектуры

Мостовое устройство PCI-PCI может быть логически разделено на следующие 4 основных компонента:

- первичный PCI интерфейс,
- вторичный PCI интерфейс,
- буферы пересылки,
- регистры конфигурации.

Блок-схема на фиг.5 изображает основные функциональные устройства.

Первичный PCI интерфейс

Первичный PCI интерфейс 71 мостового устройства PCI-PCI может выступать или как адресат, или как инициатор транзакции PCI шины. Для большинства систем первичный интерфейс должен быть подсоединен к PCI стороне моста главный процессор/PCI, которая обычно является PCI шиной с наименьшим номером в системной иерархии. Первичный интерфейс состоит из мандатных 50 сигнальных линий, определенных в "Спецификации архитектуры моста PCI-PCI", версии 1.0, и 4 необязательных линий прерывания.

Первичный PCI интерфейс реализует и PCI-устройство инициатор (ведущее) и PCI-устройство адресат (подчиненное). Когда транзакция инициируется на вторичной шине, то первичный ведущий конечный автомат, который описан в "Спецификации локальной шины PCI", версия 2.1, завершает транзакцию (запись или считывание) так, как если бы он был инициирующим устройством. Первичный PCI интерфейс в качестве PCI адресата для транзакций, которые должны быть завершены (исполнены) на вторичной шине, принимает транзакцию и передает запрос на вторичную сторону. В качестве адресата первичный PCI интерфейс использует положительное декодирование для того, чтобы востребовать PCI транзакцию, адресованную ниже моста, а затем передает транзакцию на вторичный ведущий интерфейс.

Первичный PCI интерфейс является ответственным за интерпретацию всех PCI команд, декодирование адреса и обработку ошибок.

PCI конфигурация для первичного и вторичного интерфейсов, логики маршрутизации прерывания (описана ниже), арбитража вторичной PCI шины (описан ниже) завершается (исполняется) через первичный интерфейс. Регистры пространства конфигурации поддерживают эти функции.

Вторичный PCI интерфейс

Вторичный PCI интерфейс 73 мостового устройства PCI-PCI функционирует почти так же, как и первичный интерфейс. Он включает в себя как ведущее PCI устройство, так и подчиненное PCI устройство и реализует "вторую" PCI шину с новым набором электрических PCI нагрузок для использования системой. Вторичный PCI интерфейс состоит из мандатных 49 линий. S\_RST# является выходным сигналом вместо входного на вторичной стороне.

Как подчиненный (адресат), вторичный PCI интерфейс реагирует на PCI транзакции востребования, которые не попадают во вторичное адресное пространство памяти моста или ввода/вывода, и передает их к мосту к ведущему устройству на первичной стороне. В качестве ведущего (инициатора) вторичный PCI-интерфейс является ответственным за завершение транзакций, инициированных на первичной стороне моста. Вторичный PCI интерфейс использует обратное декодирование адресных регистров моста и только передает адреса в первичном адресном пространстве через мост.

Вторичный PCI интерфейс также реализует отдельное адресное пространство для частных PCI устройств на вторичной шине, когда он игнорирует и не передает диапазон первичных адресов, определенных

во время конфигурации локальным процессором.

В специальном режиме работы вторичный PCI интерфейс выполняет положительное декодирование адреса на основании своего собственного набора адресных регистров памяти и ввода/вывода. Этот режим работы разрешается с использованием вторичного регистра разрешения декодирования SDER и имеет побочный эффект блокировки обратного декодирования стандартных адресных регистров моста на вторичном интерфейсе.

Буферы пересылки

Чтобы скрыть время задержки, имеющееся при арбитраже и захвате PCI адресата во время транзакций считывания и записи к противоположной стороне моста, мостовое устройство PCI-PCI имеет два встроенных 64-байтовых буфера 77 и 79 пересылки. Мост поддерживает и задержанные и пересылаемые транзакции.

При задержанной транзакции информация, требуемая для завершения транзакции, фиксируется, а транзакция прекращается сигналом Retry. Мост затем выполняет транзакцию от имени инициатора. Инициатор запрашивается для повторения исходной транзакции, которая прервана сигналом Retry, для завершения транзакции.

При пересылаемой транзакции ей разрешается завершиться на инициирующей шине перед завершением на шине-адресате.

Задержанная и пересылаемая транзакция более подробно обсуждаются ниже.

Мост использует два буфера пересылки буфер 77 пересылки вниз для данных, передаваемых от первичного интерфейса во вторичный интерфейс,

буфер 79 пересылки вверх для данных, передаваемых от вторичного интерфейса в первичный интерфейс.

Каждый буфер имеет соответствующие адресные/управляющие регистры для манипулирования информацией о транзакции. Регистры конфигурации

Каждое PCI устройство имеет отдельное адресное пространство конфигурации и регистры конфигурации 81. Первые 16 байт формата заголовка конфигурации моста реализуют регистры общей конфигурации моста, требуемые всеми PCI устройствами. Значение в регистре типа заголовка (только для чтения) определяет формат оставшихся 48 байт в заголовке и возвращает 01H для моста PCI-PCI.

Устройства на первичной шине могут только иметь доступ к пространству конфигурации моста PCI-PCI с помощью команд конфигурации типа 0. Устройства на вторичной PCI шине не могут обращаться к пространству конфигурации моста с помощью циклов конфигурации PCI. Регистры конфигурации хранят всю необходимую информацию об адресном декодировании, условиях ошибок и состоянии для обеих сторон моста.

Декодирование адреса

Процессор P2P предоставляет три раздельных адресных диапазона, которые используются для определения того, какие адреса памяти и ввода/вывода передаются в каком-либо направлении через мостовую часть процессора P2P. Существует два адресных диапазона для транзакций памяти и

один адресный диапазон для транзакций ввода/вывода. Мост использует регистр базового адреса и регистр границы для реализации адресного диапазона. Адресные диапазоны положительно декодируются в первичном интерфейсе с любым адресом в диапазоне, представляющем собой вторичный адрес, и поэтому способный передаваться вниз через мост. На вторичном интерфейсе адресные диапазоны декодируются обратно. Это означает, что любой адрес вне запрограммированных адресных диапазонов может быть передан вверх через мост.

Стандартное адресное декодирование мостового устройства может быть также модифицировано вторичным регистром разрешения декодирования (SDER). Биты этого регистра разрешают положительное адресное декодирование вторичным мостовым интерфейсом и блокируют основное адресное декодирование, используемое мостами PCI-PCI.

Адресное пространство ввода/вывода Мостовое устройство ввода/вывода реализует один программируемый адресный диапазон для транзакции ввода/вывода PCI. Непрерывное адресное пространство ввода/вывода определяется базовым регистром ввода/вывода (IOBR) и регистром границы ввода/вывода (IOLR) в пространстве конфигурации моста. Старшие 4 бита IOBR соответствуют AD[15:12] адресам ввода/вывода, а нижние 12 бит всегда равны 000H, задавая выравнивание по границе 4 Кбайт для адресного пространства ввода/вывода. Старшие 4 бита IOLR также соответствуют AD[15:12], а нижние 12 бит равны FFFH, задавая зернистость (размер блока), равную 4 Кбайт.

Мостовое устройство будет передавать от первичного во вторичный интерфейс транзакцию ввода/вывода, которая имеет адрес в адресном диапазоне, определенном IOBR и IOLR включительно. В этом случае первичный интерфейс выступает как PCI-адресат, а вторичный интерфейс выступает как PCI-инициатор для передаваемых через мост PCI транзакций.

Если транзакция считывания или записи ввода/вывода присутствует на вторичной шине, то мостовое устройство передает ее в первичный интерфейс, если адрес находится вне адресного пространства, заданного IOBR и IOLR. В этом случае вторичный интерфейс выступает как PCI-адресат, а первичный интерфейс служит в качестве PCI-инициатора.

Процессор P2P поддерживает только 16-битовую адресацию для транзакций ввода/вывода, и поэтому любая транзакция ввода/вывода с адресом, большим 64 Кбайт, не будет передана по какому-либо интерфейсу. Для моста предполагается, что AD [31:16] = 0000H, несмотря на то, что эти биты не реализованы в IOBR и IOLR. Мостовое устройство должно также выполнять полное 32-битное декодирование во время транзакции ввода/вывода для проверки AD [31:16] = 0000H в соответствии со "Спецификацией локальной шины PCI".

#### Режим ISA

Мостовое устройство PCI-PCI устройства P2P использует бит режима ISA в регистре управления мостом (BCR) для обеспечения осведомленности о наличии плат

ввода/вывода с шиной ISA на подчиненных PCI шинах. Режим ISA влияет только на адреса ввода/вывода в адресном диапазоне, определенном регистрами IOBR и IOLR. Когда режим ISA разрешен посредством установки бита режима ISA, мост будет отфильтровывать и не передавать транзакции ввода/вывода с адресами, находящимися в старших 768 байтах (300H) каждого естественно выровненного по границе 1 Кбайт блока. Наоборот, транзакции ввода/вывода на вторичной шине будут обратно декодировать ISA адреса и поэтому передавать транзакции ввода/вывода с адресами, находящимися в старших 768 байтах каждого естественно выровненного по границе 1 Кбайт блока.

#### Адресное пространство памяти

Мостовое устройство PCI-PCI поддерживает два различных адресных диапазона для передающихся обращений к памяти вниз от первичного во вторичный интерфейсы. Базовый регистр памяти (MBR) и регистр границы памяти (MLR) определяют один адресный диапазон, а базовый регистр памяти с предварительной выборкой (PMBR) и регистр границы памяти с предварительной выборкой (PMLR) определяют другой адресный диапазон. Адресный диапазон памяти с предварительной выборкой используется в определении, какое адресное пространство способно предварительно выбираться без побочных эффектов. Обе пары регистров определяют, когда мост будет передавать транзакции "считывание памяти", "шина считывания памяти", "коэффициент считывания из памяти", "запись в память", а также "запись в память и недостоверные данные". В случае, когда две пары регистров перекрываются, один адресный диапазон приводит к тому, что существует суммирование обоих регистров, объединенных с диапазоном предварительной выборки, имеющим приоритет перед откликом на транзакцию считывания моста.

Старшие 12 битов регистров MBR, MLR, PMBR, PMLR соответствуют адресным битам AD [31:20] адресов первичной или вторичной памяти. Для целей декодирования мост предполагает, что AD [19:0] обоих базовых регистров памяти равны 0000H и что AD [19:0] обоих регистров границы памяти равны FFFFH. Это приводит к тому, что диапазон адресов памяти, поддерживаемой мостовым устройством, выровнен по границам 1 Мбайт и имеет величину зернистости 1 Мбайт. Младшие 4 бита всех четырех регистров имеют статус "только для чтения" и возвращают 0 при считывании.

Любая транзакция памяти PCI (не ввода/вывода), присутствующая на первичной шине и попадающая в адресное пространство, определенное двумя парами регистров (MBR-MLR и PMBR-PMLR), будет передаваться вниз через мост от первичного во вторичный интерфейс. Вторичный ведущий интерфейс всегда будет использовать тот же тип PCI-команд на вторичной шине, который требуется первичным подчиненным интерфейсом на первичной шине (за исключением некоторых случаев во время "записи в память и недостоверные данные"). Все двойные адресные циклы (PCI транзакции с 64-битовыми адресами) всегда востребуются вторичным интерфейсом.



Любая транзакция памяти PCI, присутствующая на вторичной шине, которая попадает вне адресного диапазона, определенного двумя парами регистров (MBR-MLR и PMBR-PMLR), будет передаваться вверх через мост от вторичного в первичный интерфейс. Вторичный интерфейс будет передавать все двойные адресные циклы от вторичной шины в первичную шину. Двойные адресные циклы ограничены до старших 4 Гбайт 64-битового адресного пространства.

Отклик моста на транзакции памяти на любом интерфейсе может быть модифицирован следующими битами регистров из пространства конфигурации моста:

бит разрешения ведущего устройства в первичном регистре команд (PMCD),

бит разрешения памяти в первичном регистре команд (PMCD),

бит разрешения VGA в регистре управления мостом (BCR),

бит разрешения положительного декодирования вторичной памяти во вторичном регистре разрешения декодирования (SDER).

Бит разрешения положительного декодирования вторичной памяти в SDER модифицирует декодирование вторичного адреса. Он разблокирует пару регистров адресного диапазона, базовый регистр вторичной памяти (SMBR) и регистр границы памяти (SMLR), которые определяют адресное окно для транзакций, востребующих память, на вторичной шине и передающихся через мост. Транзакции декодирования и востребования работают таким же образом, как положительное декодирование на первичной шине для адресных пар MBR/MLR и PMBR/PMLR. Бит разрешения положительного декодирования вторичной памяти также блокирует обратное декодирование, выполняемое во вторичном интерфейсе, который востребует транзакции памяти с адресами вне адресного диапазона MBR/MLR и PMBR/PMLR.

Декодирование 64-битового адреса - двойной адресный цикл

Мостовое устройство поддерживает команду двойного адресного цикла для 64-битовой адресации только на вторичном интерфейсе мостового устройства. Двойные адресные циклы позволяют осуществить 64-битовую адресацию с помощью использования двух фаз PCI адресов; первая для младших 32 битов и вторая для старших 32 битов.

Мостовое устройство обычно декодирует и передает все двойные адресные циклы от вторичного в первичный интерфейс вне зависимости от адресного диапазона, определенного парами регистров MBR/MLR и PMBR/PMLR. Двойные адресные циклы не будут переданы, если бит разрешения декодирования с вычитанием для вторичной шины в SDER установлен.

Мостовое устройство будет использовать синхронизацию декодирования с вычитанием (установление уровня сигнала DEVSEL# в пятом импульсе тактового сигнала после установления уровня FRAME#) для востребования двойных адресных циклов. Это позволяет другим исполнительным устройствам на вторичной шине PCI

востребовать двойные адресные циклы до мостового устройства.

Первичный интерфейс не будет передавать двойные адресные циклы.

Механизм хранения и передачи 32 битов старшего порядка 64-битового адреса является дополнением 32-битовых адресных регистров, связанных с каналом данных из вторичного интерфейса в первичный. Эти регистры будут хранить 32 бита старшего порядка 64-битового адреса, который передается во время второй адресной фазы двойного адресного цикла. К тому же, ведущий и подчиненный конечные автоматы должны быть способны поддерживать двойной адресный цикл и DAC команду.

Ответные действия на DAC циклы во вторичном интерфейсе могут быть модифицированы с помощью следующих битов регистров в пространстве конфигурации моста

бит разрешения ведущего устройства в первичном регистре команд (PCMD),

бит разрешения памяти в первичном регистре команд (PCMD).

Бит разрешения памяти в регистре PCMD должен быть установлен для того, чтобы разрешить мосту реагировать на любой вид цикла памяти - 32 или 64 бита. Бит разрешения ведущего устройства в PCMD должен быть установлен для того, чтобы разрешить первичному интерфейсу быть ведущим для PCI транзакций.

Работа моста

Мостовое устройство процессора P2P способно передавать все типы команд работы с памятью, ввода/вывода и конфигурации от одного PCI интерфейса к другому PCI интерфейсу. Таблица 1 определяет PCI команды, поддерживаемые или не поддерживаемые мостовым устройством PCI-PCI и его двумя PCI интерфейсами. PCI команды кодируются на линиях C/BE[3:0]# на любом интерфейсе. Для предотвращения тупиковой ситуации из-за двух различных интерфейсов, мост предоставляет приоритет первичному интерфейсу, когда транзакции имеются одновременно на обоих интерфейсах.

PCI интерфейсы

Мостовое устройство P2P имеет первичный PCI интерфейс и вторичный PCI интерфейс. Когда транзакции инициируются на первичной шине и востребуются мостом, первичный интерфейс служит в качестве PCI устройства-адресата, а вторичный интерфейс служит как инициирующее устройство для истинного PCI адресата на вторичной шине. Первичная шина является инициирующей шиной, а вторичная шина является шиной-адресатом. Последовательность является обратной для транзакций, инициированных на вторичной шине. Интерфейсы определены ниже.

Первичный интерфейс

Первичный PCI интерфейс 71 мостового устройства является интерфейсом, соединенным с младшей пронумерованной PCI-шиной между двумя PCI-шинами, которые соединяет мостом P2P устройство.

Первичный PCI интерфейс должен придерживаться определения ведущего и подчиненного PCI-устройства, которые определены в "Спецификации локальной шины PCI" и "Спецификации архитектуры



PCI-моста".

Вторичный интерфейс

Вторичный PCI интерфейс 73 мостового устройства является интерфейсом, соединенным со старшей пронумерованной PCI-шиной между двумя PCI-шинами, которые соединяет мостом P2P устройство.

Вторичный PCI интерфейс должен придерживаться определения PCI-ведущего и PCI-подчиненного устройства, которые определены в "Спецификации локальной шины PCI" и "Спецификации архитектуры PCI-моста".

Буферы пересылки

Мостовое устройство PCI-PCI имеет два буфера пересылки, которые используются для задержанных транзакций и пересылаемых транзакций. Передающий вниз буфер 77 является каналом данных от первичного интерфейса во вторичный интерфейс. Передающий вверх буфер 79 является каналом данных от вторичного интерфейса в первичный интерфейс. Фиг. 5 изображает два буфера пересылки между первичным и вторичным интерфейсами.

Передающий вниз буфер пересылки используется

пересланными сигналами записи от первичной шины,

задержанными запросами записи от первичной шины,

задержанными сигналами завершения считывания, возвращающимися во вторичную шину,

задержанными сигналами завершения записи, возвращающимися во вторичную шину.

Передающий вверх буфер пересылки используется

посланными сигналами записи от вторичной шины,

задержанными запросами записи от вторичной шины,

задержанными сигналами завершения считывания, возвращающимися в первичную шину,

задержанными сигналами завершения записи, возвращающимися в первичную шину.

Пересылка записи позволяет мосту достигать своей полной потенциальной пропускной способности, в то же время скрывая задержку, связанную с передачей через мост, и задержку, связанную с захватом шины-адресата. Два набора буферов пересылки могут быть использованы одновременно.

Организация буферов пересылки

Каждый буфер пересылки может хранить 64 байта данных, организованных в 16 элементов по 4 байта каждый (16 двойных слов DWORD). Каждый буфер может хранить одну пересылаемую транзакцию записи объемом до 64 байт или

одну задержанную транзакцию завершения объемом до 64 байт, или

одну задержанную транзакцию записи объемом до 4 байт.

Связанным с каждым буфером пересылки является адресный регистр и набор тэговых битов и битов достоверности.

Мост также может сохранять один задержанный запрос считывания вне буфера пересылки.

Внутренняя адресация буферов пересылки является циклической, так что

когда транзакция поступает в пустой буфер, она должна быть немедленно передана вверх. Никакой PCI синхронизации не требуется для перемещения данных от одного элемента в буфере к другому.

Работа буфера пересылки

Оба буфера пересылки используются для того, чтобы помочь мосту достичь полной PCI пропускной способности и скрыть задержку захвата PCI шин для каждой транзакции, проходящей через мост. Бит блокировки пересылки в регистре EBCR должен быть очищен для разрешения буферам пересылать транзакции.

Природа буферов пересылки предусматривает одновременные операции от первичного во вторичный PCI интерфейсы и от вторичного в первичный PCI интерфейсы. Это означает, что транзакции, пересылаемые к противоположным интерфейсам, могут иметь место в обоих интерфейсах в одно и то же время. С момента инициирования транзакции к мосту интерфейс-адресат пытается получить статус обладания шиной-адресатом. Механизм, используемый для этого, является стандартным механизмом PCI-арбитража, используемым в первичном и вторичном интерфейсах.

В начальном состоянии по умолчанию буферы пересылки должны быть отмечены как недействительные. Любое последующее событие сброса PCI очистит все буферы, отмечая их как недействительные.

Правила упорядочения транзакций

Так как мост может обрабатывать множество транзакций, он должен поддерживать правильный порядок во избежание тупиковых условий и для увеличения пропускной способности. Таблица 2 содержит правила упорядочения множества транзакций. Первая строка содержит транзакцию, которая принята. Первая колонка является транзакцией, которая только что захвачена. Таблица указывает, что новая транзакция может пропускать предыдущую принятую транзакцию (обозначенную как DA), или новая транзакция может не пропускать предыдущую транзакцию (обозначенную как NET), или новая транзакция не может быть принята (не принимается). Не принятые транзакции могут сопровождаться сигналом "Повтор" (Retry).

Определение регистров

Ниже описываются регистры конфигурации моста PCI-PCI. Пространство конфигурации состоит из 8, 16, 24 и 32-битовых регистров, организованных в соответствии с заданным форматом. Регистры конфигурации доступны через (команды) считывания и записи конфигурации типа 0 с первичной стороны моста и через локальные операции локального процессора.

Для каждого регистра, отличного от определенных "Спецификацией локальной шины PCI" и "Спецификацией архитектуры моста PCI-PCI", описаны функциональные возможности, тип доступа (считывание/запись, считывание/очистка, только считывание) и условия установки в исходное состояние при сбросе. Как указано, требуется команда конфигурации типа 0 с первичной стороны с активным уровнем IDSEL или обращение отображенного в память локального процессора для считывания или записи в эти регистры.

Формат для регистров со смещениями до 3EH определены в "Спецификации архитектуры моста PCI-PCI", версии 1.0, и поэтому подробно не описываются регистры со смещениями более 3EH, имеющие реализацию, специфичную для процессора P2P.

Существует дополнительное требование для разрешения локальному процессору обращаться к пространству конфигурации моста. Некоторые регистры, которые имеют статус "только для считывания" из команд считывания и записи типа 0, могут быть перезаписываемыми с помощью локального процессора. Это дает возможность инициировать некоторые регистры конфигурации перед началом конфигурации PCI.

Локальный процессор считывает и записывает в пространство конфигурации моста как в отображенные в память регистры. Таблица 3 изображает регистры и связанное с ними смещение, используемые в команде конфигурирования PCI, и их отображенные в память адреса в адресном пространстве локального процессора.

Установление уровня сигнала P\_RST# с первичной стороны моста влияет на состояние большинства регистров, содержащихся в пространстве конфигурации моста. Если не указано противоположное, все биты и регистры будут возвращаться в свои заданные по умолчанию значения состояния при первичном сбросе. Исходное состояние выходного сигнала вторичного S\_RST# не влияет на состояние регистров, если не указано явно.

Как отмечено выше, биты в регистре ID продавца через регистр управления мостом - BCR придерживаются определений в "Спецификации локальной шины PCI" и поэтому не нуждаются в описании. Ниже приводится описание регистров, добавленных к "Спецификации локальной шины PCI" для реализации моста PCI-PCI, согласно настоящему изобретению. Добавленные регистры начинаются с адреса со смещением 40H, как показано в таблице 3.

Регистр управления расширенными функциями моста - EBCR

Регистр управления расширенными функциями моста используется для управления расширенными функциональными возможностями реализаций моста на основании "Спецификации архитектуры моста PCI-PCI". Он имеет биты разрешения/запрета расширенных функциональных возможностей моста (см. табл. 4a).

Первичный регистр состояния прерывания моста - PBISR

Первичный регистр состояния прерывания моста используется для оповещения локального процессора об источнике прерывания интерфейса первичного моста. К тому же, в этот регистр записывают информацию, чтобы очистить источник прерывания к устройству прерывания процессора P2P. Все биты этого регистра имеют статус "только для чтения" со стороны PCI и "считывание/очистка" со стороны локальной шины.

Биты 4: 0 являются точной копией бита 8 и битов 14:11 (соответственно) первичного регистра состояния (эти биты

устанавливаются в одно и то же время аппаратным обеспечением, но должны быть очищены независимо). Условия, которые приводят к прерыванию первичного моста, очищаются посредством записи "1" в соответствующие биты этого регистра (см. табл. 4b).

Вторичный регистр прерывания моста - SBISR

Вторичный регистр состояния прерывания моста используется для оповещения локального процессора об источнике прерывания интерфейса вторичного моста. К тому же, в этот регистр записывают информацию, чтобы очистить источник прерывания для устройства прерывания процессора P2P. Все биты этого регистра имеют статус "только для чтения" со стороны PCI и "считывание/очистка" со стороны локальной шины.

Биты 4: 0 являются точной копией бита 8 и битов 14:11 (соответственно) вторичного регистра состояния (эти биты устанавливаются в одно и то же время аппаратным обеспечением, но должны быть очищены независимо). Условия, которые приводят к прерыванию первичного моста, очищаются посредством записи "1" в соответствующие биты этого регистра (см. табл. 4c).

Вторичный регистр выбора IDSEL - SISR

Вторичный регистр выбора IDSEL управляет использованием S\_AD [20:16] в преобразованиях типа 1 к типу 0 от первичного ко вторичному интерфейсу. При операции по умолчанию специальное кодирование на первичных адресах P\_AD [15:11] приводит к установлению одного бита на вторичной адресной шине S\_AD [31:16] во время преобразования типа 1 к типу 0. Это используется для установления уровня сигнала IDSEL на устройстве, адресуемом командой конфигурации типа 0. Этот регистр разрешает использование вторичных адресных битов S\_AD[20:16] для конфигурирования приватных PCI-устройств посредством установления вторичных адресных битов S\_AD[20:16] в нули во время преобразований типа 1 к типу 0, несмотря на состояние первичных адресов P\_AD [15:11] (номер устройства в команде конфигурации типа 1).

Если любой адресный бит в S\_AD[20:16] должен быть использован для приватных вторичных PCI-устройств, то локальный процессор должен гарантировать, что соответствующий бит в регистре SISR установлен прежде, чем главное устройство пытается сконфигурировать иерархические PCI-шины (см. табл. 4d).

Вторичный регистр управления арбитражем SACR

Вторичный регистр управления арбитражем (SACR) используется для установки приоритета арбитража для каждого устройства, которое использует вторичную PCI шину. Запись значения установит арбитраж, в то время как считывание регистра возвратит запрограммированное значение. Каждому устройству задается 2-битовый приоритет. Приоритет приведен в таблице 4e.

Регистр SACR также содержит бит разрешения арбитража вторичной шины для устройства арбитража вторичной шины. Когда этот бит очищен, арбитр вторичной шины

блокируется, и мост будет возбуждать S\_REQ# на получение S\_GNTO# и выбирать S\_GNT# по S\_REQO#. Состояние по умолчанию должно быть для внутреннего устройства арбитража вторичной шины разрешено (бит разрешения арбитража вторичной шины установлен).

Регистр выбора маршрутизации прерывания PCI - PIRSR

Регистр выбора маршрутизации прерывания PCI описывается ниже со ссылкой на контроллер прерывания периферийных устройств и PCI (PPIC).

Вторичный базовый регистр ввода/вывода - SIOBR

Биты во вторичном базовом регистре ввода/вывода используются, когда вторичный PCI интерфейс разрешен для положительного кодирования. Вторичный базовый регистр ввода/вывода определяет нижний адрес (включительно) положительно декодированного адресного диапазона, который используется для определения, когда передавать транзакции ввода/вывода от вторичного интерфейса в первичный интерфейс моста. Он должен быть запрограммирован достоверным значением перед установлением вторичного регистра разрешения декодирования (SDER). Мост поддерживает только 16-битную адресацию, которая указана значением 0H в 4 младших значащих битах регистра. Старшие 4 бита программируют значением S\_AD [15:12] для нижней границы адресного диапазона. S\_AD[11:0] базового адреса всегда равны 000H, выравнивая вторичный адресный диапазон ввода/вывода по границе 4 Кбайт.

Для целей декодирования адреса, мост предполагает, что S\_AD[31:16], верхние 16 адресных битов адреса ввода/вывода равны нулю. Мост должен еще выполнять декодирование адреса полных 32 битов адреса в соответствии со "Спецификацией локальной шины PCI" и проверять, что старшие 16 битов равны 0000H.

Положительный вторичный адресный диапазон ввода/вывода (определенный SIOBR совместно с SIOLR) не подвергается воздействию в зависимости от состояния бита разрешения ISA в регистре управления мостом (BCR) (см. табл. 4f).

Вторичный регистр границы ввода/вывода - SIOLR

Биты во вторичном регистре границы ввода/вывода используются, когда вторичный PCI интерфейс разблокирован для положительного декодирования. Вторичный регистр границы ввода/вывода определяет старший адрес (включительно) положительно декодированного адресного диапазона, который используется для определения, когда передавать транзакции ввода/вывода от вторичного интерфейса в первичный интерфейс моста. Он должен быть запрограммирован достоверным значением, большим или равным значению в SIOBR, прежде, чем будут установлены бит разрешения пространства ввода/вывода в регистре команд моста и бит разрешения вторичного положительного декодирования ввода/вывода в регистре разрешения вторичного декодирования (SDER). Если значение в SIOBR больше, чем значение в SIOLR, то циклы ввода/вывода, переданные от вторичного в первичный интерфейс

(положительно декодированные), являются неопределенными. Мост поддерживает только 16-битную адресацию, которая указана значением 0H в 4 младших значащих битах регистра. Старшие 4 бита программируют S\_AD [15:12] для верхней части адресного диапазона. S\_AD [11:0] базового адреса всегда равны FFFH, задавая "зернистость" диапазона ввода/вывода, равную 4 Кбайт.

Для целей декодирования адреса, мост предполагает, что S\_AD[31:16], верхние 16 адресных битов адреса ввода/вывода равны нулю. Мост должен еще выполнять декодирование адреса полных 32 битов адреса в соответствии со "Спецификацией локальной шины PCI" и проверять, что старшие 16 битов равны 0000H.

Вторичный адресный диапазон ввода/вывода (определенный SIOBR совместно с SIOLR) не модифицируется битом разрешения ISA регистра управления мостом (см. табл. 4g).

Вторичный базовый регистр памяти - SMBR

Биты во вторичном базовом регистре памяти используются, когда вторичный PCI интерфейс моста разблокирован для положительного кодирования адреса. Они также используются для определения приватного адресного пространства на вторичной PCI шине, если бит разрешения первичного адресного пространства в регистре SDER установлен. Вторичный базовый регистр памяти определяет нижний адрес (включительно) отображенного в память адресного диапазона, который используется для определения, когда передавать транзакции из вторичного в первичный интерфейс. Вторичный базовый регистр памяти должен быть запрограммирован достоверным значением перед тем, как будет установлен бит разрешения положительного декодирования вторичной памяти в регистре SDER. Старшие 12 бит соответствуют S\_AD[31:20] 32-битовых адресов. С целью декодирования адреса мост предполагает, что S\_AD [19:0], младшие 20 адресных битов базового адреса памяти равны нулю. Это означает, что нижняя часть определенного адресного диапазона будет выровнена по границе 1 Мбайт (см. табл. 4h).

Вторичный регистр границы памяти - SMLR

Биты во вторичном регистре границы памяти используются, когда вторичный интерфейс мостового устройства разблокирован для положительного декодирования. Вторичный регистр границы памяти определяет старший адрес (включительно) отображенного в память адресного диапазона, который используется для определения, когда передавать транзакции из вторичного интерфейса в первичный интерфейс. Вторичный регистр границы памяти должен быть запрограммирован значением, большим или равным значению в SMBR прежде, чем будут установлены бит разрешения пространства памяти и бит разрешения вторичного положительного декодирования памяти. Если значение в SMLR не больше или равно значению в SMBR, когда установлены бит разрешения пространства памяти и бит разрешения вторичной памяти, то положительно декодированные транзакции

памяти от вторичного в первичный интерфейс будут неопределенными. Старшие 12 битов соответствуют S\_AD [31:20] при 32-битовой адресации. Для целей декодирования адреса, мост предполагает, что S\_AD [19:0], младшие 20 адресных битов базового адреса вторичной памяти равны FFFFFH. Это определяет "зернистость" адресного диапазона памяти, равную 1 Мбайт (см. табл. 4i).

Вторичный регистр разрешения декодирования - SDER

Вторичный регистр разрешения декодирования используется для управления функциями декодирования адреса на вторичном PCI интерфейсе мостового устройства. Бит разрешения вторичного положительного декодирования ввода/вывода, когда установлен, принуждает мост декодировать и востребовать транзакции в адресном диапазоне, определенном парой адресных регистров SIOBR/SIOLR, и передавать их через мостовое устройство. Бит разрешения вторичного положительного декодирования памяти выполняет ту же функцию, что и бит разрешения вторичного положительного декодирования ввода/вывода, но работает с адресным диапазоном SMBR/SMLR. Установка любого из этих битов блокирует все операции обратного декодирования на вторичном интерфейсе.

Бит разрешения вторичного декодирования с вычитанием предусматривает мостовое декодирование с вычитанием на вторичном интерфейсе для поддержания мостов расширения стандартной шины на первичном интерфейсе. Этот бит только разрешает декодирование с вычитанием на вторичном интерфейсе, если установлены или бит разрешения вторичного положительного декодирования ввода/вывода или бит разрешения вторичного положительного декодирования памяти.

Бит разрешения пространства приватной памяти разрешает создавать приватное пространство памяти на вторичной PCI шине. Этот бит используется совместно с регистрами SMBR/SMLR. Если этот бит установлен, то транзакции с адресами в адресном диапазоне, определяемом SMBR/SMLR, мостом игнорируются (см. табл. 4j).

Устройство преобразования адресов

Ниже следует описание устройства, которое располагается между первичной и вторичной PCI шинами и локальной шиной. Описываются режимы работы, установка и реализация интерфейса.

Обзор

Процессор P2P предоставляет интерфейс между PCI шиной и локальной шиной. Этот интерфейс содержит два устройства преобразования адресов (ATU) 43a/43b и устройство 45 передачи сообщений. Устройства ATU поддерживают как входящее, так и выходящее преобразование адресов. Первое устройство преобразования адресов называют первичным ATU 43a. Оно обеспечивает прямой доступ между первичной PCI шиной и локальной шиной. Второе устройство преобразования адресов, называемое вторичным ATU 43b, обеспечивает прямой доступ между вторичной PCI шиной и локальной шиной.

Использование двух устройств ATU, таким образом, обеспечивает значительные преимущества перед известными способами.

Во время входящих транзакций ATU преобразует PCI адреса (инициированные "хозяином" PCI-шины) в адреса локального процессора и инициирует передачу данных по локальной шине. Во время выходящих транзакций ATU преобразует адреса локального процессора в адреса PCI и инициирует передачу данных на соответствующую PCI шину.

Оба устройства преобразования адресов и устройство передачи сообщений выглядят как одно PCI-устройство на первичной PCI-шине. Эти устройства совместно являются второй PCI- функцией в многофункциональном процессоре P2P. Блок-схема устройств ATU и устройства передачи сообщений изображена на фиг. 6.

Функциональные возможности устройств ATU и устройства передачи сообщений описываются ниже. Все изображенные устройства имеют интерфейс в виде отображенных в память регистров, который является видимым или со стороны PCI-интерфейса 91, интерфейса 93 локальной шины или обоих.

Поток данных ATU

Первичное устройство ATU и вторичное устройство ATU поддерживают транзакции в обоих направлениях через P2P процессор. Первичное ATU позволяет ведущим PCI-устройствам на первичной PCI-шине инициировать транзакции к локальной шине и позволяет локальному процессору инициировать транзакции к первичной PCI-шине. Вторичное ATU выполняет ту же функцию, но на вторичной PCI-шине и для ведущих устройств вторичной PCI-шины. Транзакции, инициированные на PCI-шине и адресованные на локальную шину, называются входящими транзакциями, а транзакции, инициированные на локальной шине и адресованные на PCI-шину, называются выходящими транзакциями.

Преобразование адресов ATU

Устройства ATU реализуют схему обработки адресов способом окна для определения, какие адреса востребовать и передавать на соответствующую шину.

Первичное ATU содержит канал данных между первичной PCI-шиной и локальной шиной. Подсоединение первичного ATU таким образом дает возможность передачи данных без требования каких-либо ресурсов на вторичной PCI-шине. Вторичное ATU содержит канал данных между вторичной PCI-шиной и локальной шиной. Вторичное ATU позволяет ведущим устройствам вторичной PCI-шины непосредственно обращаться к локальной шине и памяти. Эти транзакции инициируются "хозяином" вторичной шины и не требуют какой-либо полосы частот на первичной PCI-шине.

Устройствам ATU разрешено распознавание и генерирование множества типов PCI-циклов. Таблица 5 показывает PCI-команды, поддерживаемые и входящим и выходящим устройствами ATU. Тип операции, видимый входящими устройствами ATU, определяется ведущим PCI-устройством (или на первичной или на вторичной шине), которое инициирует транзакцию. Востребование входящей транзакции зависит

от адреса, находящегося в программируемом окне входящей транзакции. Тип транзакции, используемый выходными устройствами ATU, определяется локальным адресом и фиксированной выходной схемой обработки способом окна.

Оба устройства ATU поддерживают 64-битовое расширение адресации, определенное спецификацией локальной PCI-шины. Это 64-битовое расширение адресации является только для выходящих транзакций данных (т.е. передач данных, инициированных локальным процессором).

Преобразование входящих адресов

Устройства ATU обеспечивают механизм, который позволяет "хозяевам" PCI-шины непосредственно адресовать локальную шину. Эти "хозяева" PCI-шины могут считывать или записывать отображенные в память регистры процессора P2P или пространство локальной памяти. Транзакции, когда "хозяева" PCI-шины обращаются к локальной шине, называются входящими транзакциями.

Входная транзакция включает два этапа.

1. Определение адреса.

Определение, находится ли 32-битовый PCI-адрес в адресном окне, определенном для входного ATU (первичного или вторичного).

Востребование PCI-транзакции с быстрой синхронизацией DEVSEL#.

2. Преобразование адреса.

Преобразование 32-битового PCI-адреса в 32-битовый локальный адрес.

Первичное ATU использует следующие регистры при преобразовании входящего адреса:

первичный базовый адресный регистр входного ATU,  
первичный регистр границ входного ATU,  
первичный регистр значения преобразования входного ATU.

Вторичное ATU использует следующие регистры при преобразовании входного адреса:

вторичный базовый адресный регистр входного ATU,  
вторичный регистр границ входного ATU,  
вторичный регистр значения преобразования входного ATU.

Посредством преобразования первичные входящие адреса ATU становятся первичными PCI-адресами, а вторичные входящие адреса ATU становятся вторичными PCI-адресами. В случае,

когда адрес может востребоваться и ATU и мостом, PCI-интерфейс входного ATU будет первичным.

Определение входящего адреса происходит из 32-битового PCI-адреса, регистра базового адреса и регистра границы. Алгоритм определения является следующим:

if (PCI-адрес & регистр границ == базовый регистр) then PCI адрес является востребованным входным ATU

Над входящим 32-битовым PCI-адресом производится поразрядная операция логического "И" со связанным входным регистром границы. Если результат совпадает с содержимым базового регистра, то входящий PCI-адрес является определенным в качестве попадающего во входное окно преобразования и является востребованным устройством ATU.

Когда транзакция востребована, то адрес в IAQ должен быть преобразован из 32-битового PCI-адреса в 32-битовый адрес локального процессора. Алгоритм такого преобразования следующий:

Локальный адрес = (PCI-адрес & ~регистр границ)|регистр значения

Над входящим 32-битовым PCI-адресом сначала производится поразрядная операция логического "И" с поразрядными инвертированными значениями регистра границы. Над этим результатом затем производится поразрядная операция логического "ИЛИ" с регистром значения, а результатом является локальный адрес. Этот механизм преобразования используется для всех входящих команд считывания и записи в память, исключая входящие команды считывания и записи конфигурации. Совпадение множества PCI-адресов при обращении к одному и тому же физическому локальному адресу может быть предотвращено посредством программирования входного регистра значения на границах, совпадающих со связанным регистром границы, но это осуществляется только посредством программирования прикладных задач.

Преобразование выходящего адреса

В дополнение к предоставлению механизма входного преобразования, устройства ATU обеспечивают аппаратное обеспечение, необходимое для преобразования циклов, инициированных локальным процессором, к PCI-шине. Это известно как преобразование выходящего адреса. Выходящие транзакции являются процессорными командами считывания или записи, адресованными одной из PCI-шин (первичной или вторичной). Интерфейс подчиненной локальной шины ATU будет востребовать цикл шины локального процессора и завершать цикл на PCI-шине от имени локального процессора. Первичное и вторичное устройства ATU поддерживают два различных режима выходящего преобразования

окна преобразования адреса,  
окно прямой адресации.

Устройство преобразования расширенной памяти ROM

Первичное входное ATU поддерживает один адресный диапазон (определенный парой регистров - базовый/границ), используемый для содержащейся расширенной памяти ROM. "Спецификация локальной шины PCI" указывает подробное описание форматов расширенной памяти ROM и ее использование.

Код инициализации от расширенной памяти ROM должен быть выполнен однажды главным процессором во время включения питания для инициализации соответствующего устройства. Код должен быть отброшен после его выполнения.

Входное первичное ATU будет поддерживать входное окно расширенной памяти ROM, которое работает как входное окно преобразования. Операция считывания из окон расширенной памяти ROM должна быть передана на локальную шину и в контроллер памяти. Алгоритм преобразования адреса является таким же, как и при входном преобразовании. Поддерживаются две различные ширины

(режима) ROM: 8 бит и 32 бита. Бит режима расширенной памяти ROM регистра ATUCR должен быть запрограммирован программным обеспечением для отражения физической конфигурации расширенной памяти ROM. Этот бит определяет, как ATU обращается к расширенной памяти ROM (см. ниже).

Входящее ATU выполняет следующие функции

первичное ATU определяет "попадание" в окно расширенной памяти ROM,

первичное ATU преобразует адрес (в IAQ), используя регистры ERTVR и ERLR,

для согласования с 8-битовым устройством, входное ATU выполняет четыре отдельных операции считывания на локальной шине для возвращения одного 32-битного значения на первичную PCI-шину. Каждое считывание будет состоять из 8-битного цикла с сигналом разрешения байта BE1:0#, используемым в качестве адреса байта. Каждое считывание возвращает один байт данных для аппаратного обеспечения формирования пакета (на шине AD7:0 шириной в байт), находящейся в первичном ATU.

Контроллер памяти P2P выполняет одно считывание в ответ на каждый запрос локальной шины от первичного ATU. Так как обращение происходит 8-битного устройства, то контроллер памяти будет выполнять 8-битовое считывание с 8-битовым циклом. Контроллер памяти будет возвращать данные на соответствующей шине шириной в байт (основанной на сигнале разрешения байта).

Аппаратное обеспечение формирования пакета в ATU будет возвращать целиком 32-битовое слово (от задержанной транзакции считывания) в первичную PCI-шину, когда ее 4 цикла завершены и исполнены. Аппаратное обеспечение формирования пакета ответственно за обеспечение присутствия байтов на правильной шине.

Если первичная PCI-шина требует менее 4 байтов, то первичное ATU соответственно согласует количество считанных байтов. Любые обращения, которые не являются выровненными локальной шиной, приводят к считыванию до выравнивания по границе.

Определение регистров

Каждое PCI-устройство реализует свое собственное отдельное адресное пространство конфигурации и регистры конфигурации. "Спецификация локальной шины PCI", версия 2.1, требует, чтобы пространство конфигурации имело длину 256 байт с первыми 64 байтами, придерживающимися заданного формата заголовка.

И первичное и вторичное устройства ATU программируются командами конфигурации типа 0 на первичном интерфейсе. Программирование вторичного ATU возможно посредством вторичных входных циклов конфигурации. Пространство конфигурации ATU и устройства передачи сообщений является функцией номер 1 многофункционального PCI-устройства процессора P2P.

Вне требуемого 64-байтного формата заголовка пространство конфигурации ATU и устройства передачи сообщений реализует расширенное регистровое пространство в поддержке функциональных возможностей устройств. "Спецификация локальной шины

PCI" содержит подробное описание программирования и доступа к регистровому пространству конфигурации.

Пространство конфигурации состоит из 8, 16, 24 и 32 битовых регистров, имеющих заданный формат. Таблица 6 объединяет все регистры одного адресного пространства конфигурации PCI-функции.

Каждый регистр описывается по своим функциональным возможностям, типу доступа (считывание/запись, считывание/очистка, только считывание) и условие установки в исходное состояние по умолчанию. Номер PCI-регистра для каждого регистра приводится в Таблице 6. Как указано, команда конфигурации типа 0 на первичной или вторичной шине с активным IDSEL или обращению к отображенному в память локальному процессору требует считывания или записи в эти регистры.

Биты регистров производителя ATU до регистра максимальной задержки ATU соответствует определениям "Спецификации локальной шины PCI" и поэтому ниже не описывается.

Регистр границы первичного входного ATU - PIALR

Преобразование первичного входящего адреса имеет место для передач данных от PCI-шины (исходящих от первичной PCI-шины) к локальной шине. Блок преобразования адреса преобразует PCI-адреса в адрес локального процессора. Все передачи данных преобразуются непосредственно, таким образом, "хозяин" шины, инициирующий передачи данных, может разбивать невыровненные передачи во множество передач данных. Байт разрешения может указывать, какие каналы данных являются достоверными.

Первичный входной базовый адрес преобразования специфицирован в "регистре базового адреса первичного входного ATU-PIABAR". При определении требований размера блока первичный регистр границы преобразования обеспечивает требования размера блока для первичного регистра базового адреса. Оставшиеся регистры, используемые для выполнения преобразования адреса, описаны ниже в разделе "Преобразование входного адреса".

Запрограммированное значение в регистре значения локального процессора должно быть естественно выровнено с запрограммированным значением, находящимся в регистре базового адреса. Регистр границы должен использоваться в качестве маски, таким образом младшие адресные биты, запрограммированные в регистре значения локального процессора, должны быть недостоверными. Спецификация локальной шины PCI содержит дополнительную информацию о программировании регистров базового адреса (см. табл. 7a).

Регистр значения преобразования первичного входного ATU - PIATVR

Регистр значения преобразования первичного входного ATU (PIATVR) содержит локальный адрес, используемый для преобразования адресов первичной PCI-шины. Преобразованные адреса будут направлены на локальную шину как результат преобразования адреса первичного входного ATU (см. табл. 7b).

Регистр базового адреса вторичного входного ATU - SIABAR

Регистр базового адреса вторичного входного ATU (SIABAR) определяет блок адресов памяти, где начинается окно преобразования вторичного входного ATU. Входное ATU будет декодировать и передавать запрос шины на локальную шину с преобразованным адресом для отображения в локальную память. SIABAR определяет базовый адрес и описывает размер блока необходимой памяти. Влияние на регистр базового адреса заключается в том, что когда значение FFFF. FFFFH записывается в SIABAR, то следующая операция считывания возвратит данные из регистра границы первичного входного ATU (SIALR), а не SIABAR.

Запрограммированное значение в регистре базового адреса должно совпадать с требованиями программирования PCI для выраживания адреса. Спецификация локальной шины PCI содержит дополнительную информацию по программированию регистров базовой памяти (см. табл. 7с).

Регистр границы вторичного входного ATU - SIALR

Преобразование вторичного входного адреса имеет место для передач данных, имеющих место из вторичной шины PCI на локальную шину. Блок преобразования адреса преобразует PCI-адреса в адреса локального процессора. Все передачи данных являются непосредственно преобразованными, таким образом, ведущее устройство шины, иницилирующее передачу данных, разбивает невыровненные передачи на множество передач данных. Байт дает возможность указать, какие каналы данных являются достоверными.

При определении требований размера блока вторичный регистр границы предоставляет требования к размеру блока для вторичного регистра базового адреса. Оставшиеся регистры, используемые для выполнения преобразования адреса, описаны ниже в разделе "Преобразование входящего адреса".

Запрограммированное значение в регистре значения локального процессора должно быть естественно выровнено с запрограммированным значением, находящимся в регистре базового адреса. Регистр границы должен использоваться в качестве маски, таким образом, младшие адресные биты, запрограммированные в регистре значения локального процессора, должны быть недостоверными (см. табл. 7d).

Регистр значения преобразования вторичного входного ATU - SIATVR

Регистр значения преобразования вторичного входного ATU (SIATVR) содержит локальный адрес, используемый для преобразования адресов вторичной PCI-шины в локальные адреса. Этот адрес должен быть направлен на локальную шину как результат преобразования адреса вторичного входного ATU (см. табл. 7e).

Первичный выходной регистр значения окна памяти - POMWVR

Первичный выходной регистр значения окна памяти (POMWVR) содержит первичный PCI-адрес, используемый для преобразования локальных адресов для

выходных транзакций. Этот адрес будет направляться на первичную PCI-шину в качестве результата преобразования адреса первичного выходного ATU.

Окно 0 первичной памяти располагается, начиная с адреса локального процессора 8000.000H до 807F.FFFFH с фиксированной длиной 8 Мбайт (см. табл. 7f).

Первичный выходной регистр значения окна ввода/вывода - POIOWVR

Первичный выходной регистр значения окна ввода/вывода (POIOWVR) содержит адрес считывания или записи для ввода/вывода на первичной PCI, который ATU будет преобразовывать для обращения к локальной шине.

Окно для первичного ввода/вывода располагается, начиная с локального адреса 8200.000H до 8200.FFFFH с фиксированной длиной 64 Кбайт (см. табл. 7g).

Первичный выходной регистр значения окна DAC - PODWVR

Первичный выходной регистр значения окна DAC (PODWVR) содержит адрес DAC первичной PCI, используемый для преобразования локального адреса. Этот адрес должен быть направлен на первичную PCI-шину в качестве результата преобразования адреса первичного выходного ATU. Этот регистр используется совместно с первичным регистром верхних битов 64-битного DAC.

Первичное окно DAC располагается, начиная с локального адреса 8080.000H до 80FF.FFFFH с фиксированной длиной 8 Мбайт (см. табл. 7h).

Первичный выходной регистр верхних битов 64-битного DAC - POU DR

Первичный выходной регистр верхних битов 64-битного DAC (POUDR) определяет верхние 32 бита адреса, используемого в двойном адресном цикле. Это дает возможность первичному выходному ATU непосредственно адресовать память в любом месте 64-битового адресного пространства в главном устройстве (см. табл. 7i).

Вторичный выходной регистр значения окна памяти - SOMWVR

Вторичный выходной регистр значения окна памяти (SOMWVR) содержит вторичный PCI-адрес, используемый для преобразования локальных адресов для выходящих транзакций. Этот адрес должен быть направлен на первичную PCI-шину в качестве результата преобразования адреса вторичного выходного ATU.

Окно 0 вторичной памяти располагается, начиная с адреса локального процессора 8100.000H до 817F. FFFFH с фиксированной длиной 8 Мбайт (см. табл. 7j).

Вторичный выходной регистр значения окна ввода/вывода - SOIOWVR

Вторичный выходной регистр значения окна ввода/вывода (SOIOWVR) содержит вторичный PCI адрес ввода/вывода, используемый для преобразования локальных адресов. Этот адрес должен быть направлен на первичную PCI-шину в качестве результата вторичного выходного преобразования адреса ATU.

Если бит режима загрузки вторичной PCI в регистре ATUCR установлен, то этот регистр используется для преобразования локальных адресов, которые определяют доступ к области с адресами FE00.0000H -



FFFF.FFFFH. Если этот бит очищен, то этот регистр используется для преобразования локальных адресов, которые определяют доступ к вторичному окну ввода/вывода 8201.0000H - 8201.FFFFH.

Окно вторичного ввода/вывода располагается, начиная с локального адреса 8201.0000H до 8201.FFFFH с фиксированной длиной 64 Кбайт (см. табл. 7к).

Регистр границы расширенной памяти ROM - ERLR

Регистр границы расширенной памяти ROM (ERLR) определяет размер блока адресов, который первичное ATU будет определять как адресное пространство расширенной памяти ROM. Размер блока программируется посредством записи значения в ERLR из локального процессора. Возможный диапазон программируемых значений находится в пределах от 2 Кбайт (FFFF.F800H) до 16 Мбайт (FF00.0000H) (см. табл. 7l).

Регистр значения преобразования расширенной памяти ROM - ERTVR

Регистр значения преобразования расширенной памяти ROM (ERTVR) содержит локальный адрес, который первичное ATU будет преобразовывать для обращения первичной PCI-шины. Этот адрес должен быть направлен на локальную шину в качестве результата преобразования адреса расширенной памяти ROM (см. табл. 7m).

Регистр конфигурации ATU - ATUCR

Регистр конфигурации ATU содержит биты управления для разрешения и запрета прерываний, генерируемых регистрами "дверного колокольчика". Этот регистр также управляет преобразованием выходящих адресов от первичного и вторичного выходных устройств преобразования и содержит бит определения ширины (разрядности) расширенной памяти ROM (см. табл. 7n).

Регистр состояния прерывания первичного ATU - PATUISR

Регистр состояния прерывания первичного ATU используется для сообщения локальному процессору об источнике прерывания первичного ATU или "дверного колокольчика". К тому же, в этот регистр записывают информацию для очистки источника прерывания для прерывания к устройству прерывания процессора P2P. Все биты в этом регистре имеют статус "только для чтения" с PCI и "считывание/очистка" со стороны локальной шины.

Биты 4: 0 являются точной копией бита 8 и битов 14:11 (соответственно) регистра состояния первичного ATU (эти биты устанавливаются в то же самое время аппаратным обеспечением, но должны быть очищены независимо). Биты 6:5 устанавливаются сигналом ошибки, связанным с контроллером памяти. Бит 8 предназначен для BIST программного обеспечения, а биты 10:9 предназначены для устройства передачи сообщений. Условия, которые приводят к прерыванию первичного ATU или прерывания "дверного колокольчика", очищаются посредством записи "1" в соответствующие биты этого регистра (см. табл. 7o).

Регистр состояния прерывания вторичного ATU - SATUISR

Регистр состояния прерывания первичного

ATU используется для сообщения локальному процессору об источнике прерывания первичного ATU. К тому же, в этот регистр записывают информацию для очистки источника прерывания к устройству прерывания процессора P2P. Все биты в этом регистре имеют статус "только для чтения" со стороны PCI и "считывание/очистка" со стороны локальной шины.

Условия, которые приводят к прерыванию вторичного ATU, очищаются посредством записи "1" в соответствующие биты этого регистра (см. табл. 7p).

Регистр команд вторичного ATU - SATUCMD

Биты в регистре команд вторичного ATU придерживаются определений в "Спецификации локальной шины PCI" и в большинстве случаев влияют на поведение устройства на вторичной шине PCI (см. табл. 7q).

Вторичный выходной регистр значения окна PAC - SODWVR

Вторичный выходной регистр значения окна DAC (SODWVR) содержит DAC адрес вторичной PCI, используемый для преобразования локального адреса. Этот адрес должен направляться на вторичную PCI-шину в качестве результата преобразования адреса вторичного выходного ATU. Этот регистр используется совместно с вторичным выходным регистром верхних битов 64-битового DAC.

Вторичное DAC окно располагается, начиная с локального адреса процессора 8180.000H до 81FF.FFFFH с фиксированной длиной 8 Мбайт (см. табл. 7r).

Вторичный выходной регистр верхних битов 64-битового DAC - SODUR

Вторичный выходной регистр верхних битов 64-битового DAC (SODUR) определяет верхние 32 бита адреса, используемого в течение двойного адресного цикла. Это дает возможность вторичному выходному ATU непосредственно адресовать в любом месте 64-битового главного адресного пространства (см. табл. 7s).

Первичный выходной адресный регистр цикла конфигурации POCCAR

Первичный выходной адресный регистр цикла конфигурации используется для хранения 32-битного адреса PCI-конфигурации. Локальный процессор записывает адрес циклов PCI-конфигурации, который затем дает возможность записи и считывания первичной выходной конфигурации. Локальный процессор затем выполняет операцию считывания или записи в Первичный выходной адресный регистр данных цикла конфигурации для инициирования цикла конфигурации на первичной PCI-шине (см. табл. 7s').

Вторичный выходной регистр цикла конфигурации - SOCCAR

Вторичный выходной адресный регистр цикла конфигурации используется для хранения 32-битного адреса для цикла PCI-конфигурации. Локальный процессор записывает адрес цикла PCI-конфигурации, который затем дает возможность записи и считывания вторичной выходной конфигурации. Локальный процессор затем выполняет операцию считывания или записи во вторичный выходной адресный регистр данных цикла конфигурации для



инициирования цикла конфигурации на вторичной PCI-шине (см. табл. 7u).

Первичный выходной регистр данных цикла конфигурации POCDDR

Первичный выходной регистр данных цикла конфигурации используется для инициирования считывания или записи конфигурации на первичной PCI-шине. Регистр является логическим, а не физическим, означая, что существует адрес, а не регистр. Локальный процессор будет считывать или записывать адрес отображенных в память регистров данных для инициирования цикла конфигурации на PCI шине с адресом, находящимся в POCDDR. Для записи конфигурации данные захватываются с локальной шины и передаются непосредственно в ATU ODQ. Для считывания данные возвращаются прямо из ATU ODQ в локальный процессор и реально никогда не вводятся в регистр данных (который физически не существует).

POCCAR пригоден для использования только из адресного пространства локального процессора и проявляется как зарезервированное значение в пространстве конфигурации ATU.

Вторичный выходной регистр данных цикла конфигурации - SOCCDR

Вторичный выходной регистр данных цикла конфигурации используется для инициирования считывания или записи конфигурации на вторичной PCI-шине. Регистр является логическим, а не физическим, означая, что существует адрес, а не регистр. Локальный процессор будет считывать или записывать адреса отображенных в память регистров данных для инициирования цикла конфигурации на PCI шине с адресом, находящимся в SOCCAR. Для записи конфигурации данные захватываются с локальной шины и передаются непосредственно в ATU ODQ. Для операции считывания данные возвращаются прямо из ATU ODQ в локальный процессор и реально никогда не вводятся в регистр данных (который физически не существует).

SOCCAR пригоден для использования только из адресного пространства локального процессора и проявляется как зарезервированное значение в пространстве конфигурации ATU.

Устройство передачи сообщений

Ниже следует описание устройства передачи сообщений процессора P2P. Устройство передачи сообщений тесно связано с устройством преобразования первичного адреса (PATU), описанного выше.

Обзор

Устройство передачи сообщений обеспечивает механизм для данных, которые должны быть переданы между системой PCI и локальным процессором и оповещения соответствующей системы о поступлениях новых данных через прерывание. Устройство передачи сообщений может быть использовано для посылки и приема сообщений.

Устройство передачи сообщений имеет пять различных механизмов передачи сообщений. Каждый позволяет главному процессору или внешнему исполнительному PCI-устройству и процессору P2P обмениваться посредством передачи

сообщений и формирования прерывания.

Этими пятью механизмами являются

регистры сообщений,  
регистры "дверного колокольчика",  
циклические очереди,  
индексные регистры,  
регистры APIC.

Регистры сообщений позволяют процессору P2P и внешним исполнительным PCI-устройствам обмениваться посредством передачи сообщений в один из четырех 32-битовых регистров сообщений. В этом контексте сообщением является любое 32-битовое значение. Регистры сообщений объединяют аспекты регистров электронной почты и регистров "дверного колокольчика". Операции записи в регистры сообщений необязательно могут вызывать прерывания.

Регистры "дверного колокольчика" позволяют процессору P2P устанавливать уровень сигналов PCI-прерывания и позволяют внешним исполнительным PCI-устройствам формировать прерывание к локальному процессору.

Циклические очереди поддерживают схему передачи сообщений, которая использует 4 циклических очереди.

Индексные регистры поддерживают схему передачи сообщений, которая использует часть локальной памяти процессора P2P для реализации большого набора регистров сообщений.

Регистры APIC поддерживают интерфейсное устройство шины APIC посредством предоставления внешнего PCI-интерфейса для обращающихся регистров APIC.

Теория работы

Устройство передачи сообщений имеет пять уникальных механизмов передачи сообщений.

Регистры сообщений аналогичны объединению электронной почты и регистров "дверного колокольчика".

Регистры "дверного колокольчика" поддерживают и аппаратные и программные прерывания. Регистры "дверного колокольчика" имеют две цели

формируют прерывание при записи, поддерживают состояние прерывания формируемых другими механизмами передачи сообщений.

Устройство передачи сообщений использует первые 4 Кбайт первичного входного окна преобразования в первичном устройстве преобразования адреса (PATU). Адрес первичного входного окна преобразования находится в регистре базового адреса первичного входного ATU.

Таблица 8 представляет собой обзор пяти механизмов передачи сообщений в устройстве передачи сообщений.

Регистры сообщений

Сообщения могут быть посланы и приняты процессором P2P посредством использования регистров сообщений. При записи регистры сообщений могут вызывать прерывания, которые должны быть сформированы к локальному процессору или сигналам прерывания PCI. Входящие сообщения пересылаются главным процессором и принимаются P2P процессором. Выходные сообщения посылаются процессором P2P и принимаются главным процессором.

Состояние прерывания для выходящих сообщений записывается в выходной регистр "дверного колокольчика". Состояние прерывания для входящих сообщений записывается во входной регистр "дверного колокольчика".

#### Выходящие сообщения

Когда в регистр выходящего сообщения записывается информация локальным процессором, то прерывание может быть сформировано на линиях прерывания P\_INTA#, P\_INTB#, P\_INTC# или P\_INTD#. Какая из линий прерывания используется, определяется значением регистра линии прерывания ATU.

Прерывание PCI записывается в выходной регистр "дверного колокольчика". Прерывание вызывает выходящее сообщение, которое должно быть установлено в выходном регистре "дверного колокольчика". Им является бит считывания/очистки, который устанавливается аппаратным обеспечением устройства передачи сообщений и очищается программным обеспечением.

Прерывание очищается, когда внешнее исполнительное PCI-устройство записывает значение 1 бит выходящего сообщения в регистре выходного "дверного колокольчика" для очистки этого бита.

#### Входящие сообщения

Когда в регистр входящего сообщения происходит запись внешним исполнительным PCI-устройством, прерывание может быть сформировано к локальному процессору. Прерывание может быть замаскировано битами маски во входном регистре маскирования "дверного колокольчика".

Прерывание локального процессора записывается во входной регистр "дверного колокольчика". Прерывание вызывает установку бита входящего сообщения во входном регистре "дверного колокольчика". Им является бит считывания/очистки, который устанавливается аппаратным обеспечением устройства передачи сообщений и очищается программным обеспечением.

Прерывание очищается, когда локальный процессор записывает значение 1 в бит прерывания входящего сообщения во входном регистре "дверного колокольчика".

#### Регистры "дверного колокольчика"

Существует два регистра "дверного колокольчика": входной регистр "дверного колокольчика" и выходной регистр "дверного колокольчика". Входной регистр "дверного колокольчика" позволяет внешнему исполнительному PCI-устройству формировать прерывание к локальному процессору. Выходной регистр "дверного колокольчика" позволяет локальному процессору формировать PCI-прерывание. Оба регистра хранят объединение прерывания, сформированного аппаратным обеспечением, и прерывания, сформированного программным обеспечением. Они содержат состояние прерывания от других механизмов устройства передачи сообщений, и они также позволяют программному обеспечению устанавливать отдельные биты для вызывания прерывания.

Выходные регистры "дверного колокольчика"

Когда во входной регистр "дверного колокольчика" информацию записывает локальный процессор, то прерывание может

быть сформировано на линиях прерывания P\_INTA#, P\_INTB#, P\_INTC# или P\_INTD#. Прерывание формируется, если любой из битов в регистре "дверного колокольчика" записан равным значению 1. Запись значения 0 в любой из битов не изменяет значение этого бита и не вызывает формирование прерывания. Когда бит установлен в выходном регистре "дверного колокольчика", он не может быть очищен локальным процессором.

Какая из линий прерывания используется, определяется значением регистров линии прерывания ATU.

Прерывание может быть замаскировано битами маски в регистре маски выходного "дверного колокольчика". Если бит маски установлен для конкретного бита, никакое прерывание не формируется для этого бита. Регистр маски выходного "дверного колокольчика" вызывает только формирование прерывания и никакое значение не записывается в регистр выходного "дверного колокольчика".

Прерывание очищается, когда внешнее исполнительное PCI-устройство записывает значение 1 в биты выходного регистра "дверного колокольчика", которые устанавливаются. Запись значения 0 в любой бит не изменяет значения этого бита и не очищает прерывание.

В общем, локальный процессор формирует прерывание посредством установки битов в выходном регистре "дверного колокольчика" и внешнее исполнительное PCI-устройство очищает прерывание также с помощью установки битов в том же самом регистре.

Входные регистры "дверного колокольчика"

Когда во входной регистр "дверного колокольчика" записывается информация внешним исполнительным PCI-устройством, то прерывание может быть сформировано к локальному процессору. Прерывание формируется, если любой из битов в регистре "дверного колокольчика" записан равным значению 1. Запись значения 0 в любой из битов не изменяет значение этого бита и не вызывает формирование прерывания. Когда бит установлен в выходном регистре "дверного колокольчика", он не может быть очищен любым внешним исполнительным PCI-устройством.

Прерывание может быть замаскировано битами маски в регистре маски входного "дверного колокольчика". Если бит маски установлен для конкретного бита, никакое прерывание не формируется для этого бита. Регистр маски входного "дверного колокольчика" вызывает только формирование прерывания и никакое значение не записывается во входной регистр "дверного колокольчика".

Один бит входного регистра "дверного колокольчика" резервируется для прерывания NMI. Это прерывание не может быть замаскировано регистром маски входного "дверного колокольчика".

Прерывание очищается, когда локальный процессор записывает значение 1 в биты входного регистра "дверного колокольчика", которые установлены. Запись значения 0 в любой бит не изменяет значения этого бита и не очищает прерывание.

# Циклические очереди

Устройство передачи сообщений реализует 4 циклических очереди. Существует 2 входных очереди и 2 выходных очереди. В этом случае "входящий" и "выходящий" относится к направлению потока сообщений. Входящие сообщения являются или новыми сообщениями, передаваемыми другими процессорами для локального процессора для обработки, или являются пустыми или свободными сообщениями, которые могут быть повторно использованы другими процессорами. Выходящие сообщения являются или пересылаемыми сообщениями локальным процессором для других процессоров для обработки или являются свободными сообщениями, которые могут быть повторно использованы локальным процессором.

Четыре циклических очереди используются для передачи сообщений следующим образом. Две входных очереди используются для хранения входящих сообщений, а выходные очереди используются для хранения исходящих сообщений. Одна из входных очередей является назначенной свободной очередью, и она содержит входящие свободные сообщения. Другая входная очередь является назначенной очередью для пересылки, и она содержит входные пересылаемые сообщения. Аналогично, одна из выходных очередей является назначенной свободной очередью, а другая выходная очередь является назначенной очередью для пересылки.

Устройство арбитража вторичной PCI-шины

Ниже следует описание устройства 53 арбитража PCI-шины. Режимы работы, установка и реализация арбитража описаны ниже.

## Обзор

Локальная шина PCI требует центральных ресурсов разрешения конфликтов для каждой отдельной PCI-шины в системном оборудовании. PCI использует концепцию доступа, основанную на арбитраже, вместо традиционного подхода с выделением промежутка времени. Для каждого устройства на PCI-шине будет осуществляться арбитраж за обладание шиной для каждого момента времени, когда устройство, функционирующее как ведущее устройство шины, требует обращения к шине.

Арбитраж PCI использует простой протокол квитирования с установлением связей REQ# и GNT#. Когда устройство запрашивает шину, оно устанавливает выходной уровень сигнала REQ#. Устройство арбитража 53 будет позволять доступ запрашивающего исполнительного устройства к шине с помощью установления уровня сигнала GNT# на входе этого исполнительного устройства.

Арбитраж PCI является "скрытой" схемой арбитража, когда последовательность арбитража имеет место в фоновом режиме, в то время как некоторые другие "хозяева" шины в это время осуществляют управление шиной. Это имеет преимущество, заключающееся в том, что не занимается какая-либо полоса PCI-шины на накладные расходы арбитража шины. Единственным требованием, предъявляемым PCI-арбитру, является то, что должен быть реализован

алгоритм справедливого арбитража. Выбранный алгоритм арбитража должен гарантировать, что в любой момент времени на отдельной шине PCI будет не более одного активного сигнала GNT#.

## Теория работы

Арбитр 53 вторичной шины поддерживает до 6 вторичных "хозяев" шины плюс сам интерфейс вторичной шины. Каждый запрос должен быть запрограммирован на один из трех уровней приоритета или заблокирован. Отображенный в память регистр управления запрограммированный программным обеспечением прикладной задачи, устанавливает приоритеты для каждого из "хозяев" шины. Каждый уровень приоритета управляется циклическим способом. Циклический способ определяется как механизм, в котором каждое устройство будет по очереди выступать в качестве ведущего устройства шины. Последовательность перемещается круговым образом. Следующий возможный "хозяин" шины должен быть непосредственно перед текущим "хозяином" шины, а предыдущий "хозяин" шины должен быть непосредственно позади текущего "хозяина" шины.

Круговая схема арбитража поддерживает три уровня циклического арбитража. Эти три уровня определяют низкий, средний и высокий приоритет. Использование механизма циклического арбитража гарантирует, что для каждого уровня приоритета должен быть "победитель". Чтобы реализовать концепцию справедливости, резервируется интервал времени для "победителя" каждого уровня приоритета (за исключением наивысшего) в следующем наивысшем приоритете. Если "победителю" уровня приоритета шина не предоставляется во время этой конкретной последовательности арбитража, он должен быть продвинут к следующему наивысшему уровню приоритета. Когда шина предоставлена, он возвращается к своему запрограммированному приоритету. Резервируя этот интервал времени, алгоритм гарантирует справедливость, разрешая запросам с более низким приоритетом продвигаться посредством механизма приоритета до состояния устройства с наивысшим приоритетом и предоставляя шину в следующий интервал времени.

Арбитр связывается со всеми запрашивающими исполнительными устройствами на шине через протокол REQ# - GNT#. "Хозяин" шины будет устанавливать уровень своего выходного сигнала REQ# и ожидать установления уровня сигнала GNT# на своем входе. Исполнительному устройству может быть предоставлена шина, в то время как предыдущий "обладатель" шины еще осуществляет управление. Арбитр только является ответственным за решение, какое PCI устройство захватывает шину следующим. Каждое отдельное PCI устройство является ответственным за определение, когда в действительности шина становится свободной и ему разрешено начать свое обращение к шине.

Арбитр 53 вторичной шины может быть заблокирован посредством программирования интерфейса для возможности реализации алгоритма арбитража пользователя. Когда такая блокировка имеет место, один набор

сигналов REQ# - GNT# будет служить в качестве сигналов арбитража для вторичного PCI-интерфейса P2P.

Механизм приоритета

Механизм приоритета является программируемым или кодом BIOS (базовой системы ввода/вывода) или программным обеспечением прикладной задачи. Приоритет отдельного "хозяина" шины будет определять, какой уровень будет присвоен устройству в круговой схеме. Этот приоритет определяет, должно ли иметь устройство начальный приоритет или самый низкий приоритет. Если прикладная задача программирует устройству низкий приоритет, то устройство может быть продвинуто до среднего, а затем до высокого приоритета до тех пор, пока ему не будет предоставлена локальная шина. Как только она предоставлена, устройство должно быть установлено в начальное состояние запрограммированного приоритета и может повторно инициировать арбитраж. Дополнительные подробности, необходимые для реализации подходящего устройства арбитража вторичной PCI-шины, могут быть найдены специалистами в области, к которой относится изобретение.

Контроллер DMA

Ниже приводится описание интегрированного контроллера прямого доступа к памяти (DMA), применяемого в настоящем изобретении. Режимы работы, установка, внешний интерфейс и реализация контроллера DMA описаны ниже.

Обзор

Контроллер DMA предоставляет возможность высокоскоростной с малой задержкой передачи данных. Контроллер DMA оптимизирует блочную передачу данных между шиной PCI и памятью локального процессора. DMA является инициатором на PCI-шине с возможностью PCI-пакета обеспечивать максимальную пропускную способность 132 Мбайт/сек на частоте 33 МГц.

Аппаратное обеспечение контроллера DMA ответственно за выполнение передачи данных и за обеспечение программирования интерфейса. Особенности контроллера DMA

три независимых канала, использование интерфейса контроллера памяти P2P,

2<sup>32</sup> - диапазон адресов на интерфейсе локального процессора,

2<sup>64</sup> - диапазон адресов на первичном и вторичном PCI-интерфейсах посредством использования двойного адресного цикла (DAC),

независимые PCI-интерфейсы для первичной и вторичной PCI-шин,

аппаратная поддержка для передачи невыровненных данных для PCI-шины и локальной шины локального процессора,

полная 132 Мбайт/сек пакетная поддержка для PCI-шины и локальной шины P2P,

прямая адресация к PCI-шине и с PCI-шины,

возможность полного программирования от локального процессора,

поддержка автоматического сцепления данных для сбора и распределения блоков данных,

поддержка режима запроса для внешних устройств на 0-м канале DMA.

Фиг. 7 изображает подключение каналов

DMA к PCI-шинам.

Теория работы

Контроллер DMA 51a и 51b предоставляет три канала высокоскоростных передач от PCI в память. Каналы 0 и 1 передают блоки данных между первичной PCI-шиной и локальной памятью локального процессора. Канал 2 передает блоки данных между вторичной PCI-шиной и локальной памятью локального процессора. Все каналы идентичны, за исключением канала 0. Он имеет дополнительную поддержку для передач режима запроса. Каждый канал имеет интерфейс PCI-шины и интерфейс локальной шины локального процессора.

Каждый DMA-канал использует непосредственную адресацию и для PCI-шины и для локальной шины локального процессора. Он поддерживает передачи данных в и из полного 64-битового адресного диапазона PCI-шины. Это включает 64-битовую адресацию с использованием DAC PCI команд. Канал имеет специальный регистр, который содержит верхние 32 адресных бита для 64-битового адреса. Каналы DMA не поддерживают передачи данных, которые пересекают границу 32-битового адреса.

Дополнительные подробности, необходимые для реализации подходящего механизма DMA для использования в настоящем изобретении, могут быть обнаружены специалистами в области, к которой относится изобретение.

Контроллер памяти

Ниже приводится описание интегрированного контроллера памяти 47, применяемого в настоящем изобретении. Режимы работы, установка, внешний интерфейс и реализация контроллера памяти описаны ниже.

Обзор

Процессор P2P объединяет контроллер 47 основной памяти для обеспечения прямого интерфейса между процессором P2P и системной памятью 33. Контроллер памяти поддерживает

до 256 Мбайт 32-битового или 36-битового (32-битные данные памяти плюс 4 бита четности) DRAM (динамическую память со случайным доступом),

память DRAM с чередованием адресов и без чередования,

память DRAM типа fast-page (FPM), быстрая страничная,

память DRAM типа extended data out

(EDO), расширенная выдача данных,

память DRAM типа burst extended data out (BEDO), пакетная расширенная выдача данных,

два независимых банка для SRAM/ROM, до 16 Мбайт (на один банк) 8-битной или 32-битной SRAM/ROM.

Контроллер памяти генерирует строб выборки адреса по строке (RAS#), строб выборки адреса по колонке (CAS#), сигнал разрешения записи (WE#) и 12-битовые мультиплексированные адреса (MA[11:0]) для массива DRAM. Для памяти DRAM с чередованием адресов формируются сигналы разрешения фиксации адреса и данных (DAIE#) и (LEAF#) DRAM.

Контроллер памяти поддерживает два банка SRAM, ROM или и памяти на ЭСППЗУ с параллельным стиранием. Каждый банк

поддерживает от 64 Кбайт до 16 Мбайт памяти. Каждый банк может быть независимо сконфигурирован для 8-битовой или 32-битовой разрядности памяти. Контроллером памяти выдаются сигналы выбора кристалла (CE#), разрешения записи в память (MWE#) и увеличения адреса пакета для SRAM/ROM.

Обзор контроллера памяти, интегрированного в процессор P2P, приводится на фиг. 8.

Теория работы

Контроллер памяти 47 оптимально преобразует протокол импульсного обращения "хозяев" локальной шины в протокол доступа, поддерживаемый адресуемой памятью. Устройство декодирования 101 адреса декодирует адреса локальной шины, присутствующие на внутренней шине 41 адресов/данных, и формирует правильные сигналы управления и адреса к массиву памяти 33, подключенного к контроллеру памяти. Пакетные обращения, формируемые ведущим устройством локальной шины, выдают первый адрес. Контроллер памяти обеспечивает увеличение адреса, который подается на массив памяти по линиям MA[11:0]. Адрес увеличивается до тех пор, пока или завершится цикл ведущего устройства локальной шины, обозначенный установлением сигнала BLAST#, или имеет место ошибка четности на локальной шине (для циклов считывания DRAM). Максимальный размер пакета для одного цикла передачи данных составляет 2 Кбайт. Ведущие устройства локальной шины ответственны за сохранение отслеживания за увеличением счетчика пакета и окончание передачи данных при достижении границы адреса в 2 Кбайт.

Адрес, присутствующий на шине MA[11:0] 103, зависит от типа адресуемого банка памяти. Для DRAM MA[11:0] обеспечивает адрес с мультиплексированными строками и колонками. Адрес колонки увеличивается до ближайшей 2-Кбайтной границы адреса. Для банков памяти SRAM и FLASH/ROM шина MA[11:0] основана на адресе, присутствующем на линиях AD[13:2] в адресной фазе. Для пакетных данных счетчик пакетов 105 будет увеличивать адрес до ближайшей 2-Кбайтной границы адреса.

Контроллер памяти формирует ряд режимов ожидания в качестве запрограммированного значения в регистры 107 контроллера памяти для управления сигналами, подаваемыми на массивы памяти. К тому же, с помощью генератора 109 режимов ожидания сигнал WAIT# выдается ведущим устройством локальной шины (за исключением локального процессора) для указания дополнительных режимов ожидания, требуемых во время обращения к памяти.

Устройство 111 проверки и формирования четности данных шириной в байт может быть доступно для массивов DRAM. Проверка на четность выдает сигнал ошибки памяти при контроле на четность при обнаружении ошибки четности. Адрес сбойного слова фиксируется в регистре.

Контроллер памяти предоставляет монитор 113 шины для обнаружения адресных диапазонов, которые не возвращают внешний сигнал RDYRCV#. Этот механизм предназначен для определения

обращений к неопределенным адресным диапазонам. При обнаружении ошибки генератор формирования режима ожидания формирует внутренний сигнал LRDYRCV# для завершения обращения к шине и необязательно формирует сигнал сбоя шины.

Дополнительные подробности, необходимые для реализации подходящего контроллера памяти для использования в настоящем изобретении, должны быть очевидны специалистам в области, к которой относится изобретение.

Контроллер прерывания PCI и периферийных устройств

Ниже приводится описание контроллера прерывания процессора P2P. Режимы работы, установка, интерфейс внешней памяти и реализация прерываний описаны ниже.

Обзор

Контроллер прерывания PCI и периферийных устройств (PPIC) 67 предоставляет возможность формирования прерываний на локальный процессор и шину PCI. P2P процессор содержит несколько периферийных устройств, которые могут генерировать прерывание к локальному процессору. Эти устройства включают

канал DMA 0,  
первичное ATU,  
канал DMA 1,  
вторичное ATU,  
канал DMA 2,

интерфейсное устройство шины I<sup>2</sup>C,  
интерфейс первичного моста,  
интерфейсное устройство шины APIC,  
интерфейс вторичного моста,  
устройство передачи сообщений.

Дополнительно к внутренним устройствам внешние устройства также могут формировать прерывания к локальному процессору. Внешние устройства могут генерировать прерывания по линиям XINT7:0# и линии NMI#.

Контроллер прерывания PCI и периферийных устройств предоставляет возможность непосредственного прерывания PCI. Логические схемы маршрутизации дают возможность под управлением программного обеспечения возможность перехвата внешних прерываний вторичной PCI и направления их на линии прерывания первичной PCI.

"Руководство пользователя микропроцессора 1960 jx" содержит дополнительные подробности прерывания локального процессора и механизма приоритетов прерывания. Руководство пользователя также предоставляет полное объяснение различных режимов работы контроллера прерывания локального процессора.

Теория работы

Контроллер прерывания PCI и периферийных устройств имеет две функции управление прерываниями внутренних периферийных устройств, маршрутизация прерывания PCI.

Механизм управления прерываниями периферийных устройств объединяет несколько источников прерывания для заданного набора периферийных устройств в одно прерывание, направляемое к локальному процессору. Для того чтобы обеспечить выполнение программ со знанием источника прерывания, существует

отображенный в память регистр состояния, который описывает источник прерывания. Все прерывания от периферийных устройств разрешаются индивидуально от соответствующих регистров управления периферийными устройствами.

Механизм маршрутизации PCI прерываний позволяет программному обеспечению главного устройства (или программному обеспечению локального процессора) маршрутизировать PCI прерывания или к локальному процессору или выходные линии P\_INTA#, P\_INTB#, P\_INTC#, P\_INTD#. Этот механизм маршрутизации управляется посредством отображенного в память регистра состояния, доступного из пространства конфигурации первичного PCI-моста или локальной шины процессора P2P.

Прерывания локального процессора

Контроллер прерывания на локальном процессоре имеет 8 внешних линий прерывания и одну линию немаскируемого прерывания для обнаружения внешних запросов на прерывание. Восемь внешних линий прерывания могут быть сконфигурированы для одного из трех режимов: назначения, расширенного и смешанного. В режиме назначения линии могут быть индивидуально отображены на вектора прерывания. В расширенном режиме линии могут быть интерпретированы как битовое поле, которое может представлять собой вектор прерывания. В этом режиме с помощью линий прерывания могут быть непосредственно запрошены до 240 векторов. В смешанном режиме пять линий работают в расширенном режиме и могут запрашивать 32 различных прерывания, а три линии работают в режиме назначения.

Девять линий прерывания локального процессора имеют следующие определения и опции программирования

XINT7: 0# Внешнее прерывание (входное) - по этим линиям запрашиваются прерывания. Линии конфигурируют программно для трех режимов: назначения, расширенного и смешанного. Каждая линия может быть запрограммирована как вход обнаружения фронта или вход обнаружения уровня сигнала. К тому же, для этих линий может быть выбран режим устранения биений (колебаний) под управлением программы.

NMI# Немаскируемое прерывание (входное) - вызывает появление немаскируемого прерывания. NMI является распознаваемым прерыванием с наивысшим приоритетом. Линия NMI# является входом, активизируемым по фронту. Режим устранения колебания для NMI# может быть выбран под управлением программы. Эта линия является внутренне синхронизированной.

Для правильной работы процессора P2P линии внешнего прерывания локального процессора должны быть запрограммированы только для работы в прямом режиме, прерываний, чувствительных к уровню, и режимах быстрой выборки. Это осуществляется через регистр управления прерываниями (ICON) в отображенном в память регистровом пространстве локального процессора. "Руководство пользователя микропроцессора 1960 Jx" содержит все подробности по программированию

контроллера прерываний локального процессора.

Использование механизма прерываний процессора P2P основано на конфигурации контроллера прерываний локального процессора и бите выбора XINT в регистре выбора маршрутизации прерывания PCI. Таблица 9 описывает рабочие режимы и функциональные возможности, разрешенные контроллером прерываний локального процессора, а таблица 10 описывает использование бита выбора XINT.

Операционные блоки

Контроллер прерываний от PCI и периферийных устройств обеспечивает подсоединения к локальному процессору. Эти подсоединения приведены на фиг. 9.

Маршрутизация прерываний PCI

Четыре входа прерывания PCI могут быть маршрутизированы мультиплексорами 121 или ко входам прерываний локального процессора или к выходным линиям прерываний PCI. Маршрутизация входных сигналов прерывания управляется битом выбора XINT регистра выбора маршрутизации прерывания, который показан в таблице 10.

Как указано ранее, XINT0# - XINT3# локального процессора должны быть запрограммированы так, чтобы воспринимать уровень, для приспособления прерываний PCI. К тому же, логические схемы, внешние к входу локального процессора, должны управлять неактивным уровнем ("1"), когда бит выбора XINT установлен.

Маршрутизация прерывания внутренней периферии

Входы прерываний XINT6#, XINT7# и NMI# в локальном процессоре принимают входные сигналы от множества внутренних источников прерываний. Существует один внутренний фиксатор перед каждым из этих трех входов, который обеспечивает необходимое мультиплексирование различных источников прерываний. Прикладное программное обеспечение может определить, какое периферийное устройство вызвало прерывание, посредством считывания соответствующего фиксатора прерывания. Более подробно о точной причине прерывания можно определить при считывании состояния от периферийного устройства.

Прерывание XINT6# локального процессора принимает прерывания от внешней линии и трех каналов DMA. Каждое прерывание канала DMA является DMA-прерыванием окончания передачи или DMA-прерыванием окончания цепочки. Фиксатор прерывания XINT6 123 принимает входные сигналы прерывания от каналов DMA, а также от внешней линии XINT6# . Достоверное прерывание от любого из этих источников устанавливает бит в фиксаторе и выдает чувствительное к уровню прерывание на вход XINT6# локального процессора. Фиксатор прерывания должен продолжить подавать активный входной сигнал с низким уровнем на вход прерывания локального процессора до тех пор, пока он присутствует в фиксаторе. Фиксатор прерывания XINT6# считывается посредством регистра состояния прерывания XINT6. Фиксатор прерывания XINT6 очищается с помощью очистки источника прерывания во внутреннем периферийном устройстве.

Источники прерывания устройства, которые подают входные сигналы на фиксатор прерывания XINT6#, подробно описываются в таблице 11.

Прерывание XINT7# локального процессора принимает прерывания от внешней линии, интерфейсного устройства шины APIC, интерфейсного устройства шины I<sup>2</sup>C, первичного ATU и устройства передачи сообщений. Фиксатор прерывания XINT7 125 принимает один входной сигнал прерывания от каждого из четырех указанных выше устройств, а также от внешней линии XINT7#. Достоверное прерывание от любого из этих источников устанавливает бит в фиксаторе и выдает чувствительное к уровню прерывания на вход XINT7# локального процессора. Фиксатор прерывания должен продолжить подавать активный входной сигнал с низким уровнем на вход прерывания локального процессора до тех пор, пока он присутствует в фиксаторе. Фиксатор прерывания XINT7# считывается посредством регистра состояния прерывания XINT7. Фиксатор прерывания XINT7 очищается с помощью очистки источника прерывания во внутреннем периферийном устройстве.

Источники прерывания устройств, которые подают входные сигналы на фиксатор прерывания XINT7#, подробно описываются в таблице 12.

Немаскируемое прерывание (NMI) на локальном процессоре принимает прерывание от внешней линии, первичного и вторичного устройств ATU, первичного и вторичного мостовых интерфейсов, локального процессора и каждого из трех каналов DMA. Каждое из этих 8 прерываний представляет собой условие ошибки в периферийном устройстве. Фиксатор прерывания NMI 127 принимает один входной сигнал прерывания от каждого из восьми указанных выше устройств и внешней линии NMI#. Достоверное прерывание от любого из этих источников устанавливает бит в фиксаторе и выдает срабатывающее по фронту прерывание на вход NMI# локального процессора. Фиксатор прерывания NMI считывается посредством регистра состояния NMI прерывания. Фиксатор прерывания NMI очищается с помощью очистки источника прерывания во внутреннем периферийном устройстве.

Источники прерывания устройств, которые подают входные сигналы на фиксатор прерывания NMI, подробно описываются в таблице 13.

Регистр выбора программы прерывания PCI, регистр состояния прерывания XINT6, регистр состояния прерывания XINT7 и регистр состояния прерывания NMI описаны ниже.

Интерфейс внешнего прерывания процессора P2P

Входной интерфейс внешнего прерывания процессора P2P состоит из линий, представленных в табл. 14.

Прерывание "дверного колокольчика" выходящего PCI

Процессор P2P имеет возможность формировать прерывания на любой из линий прерывания первичного PCI. Это осуществляется установкой бита в регистре порта "дверного колокольчика" в первичном ATU. Биты 0 - 3 соответствуют P\_INTA# -

P\_INTD# соответственно. Установка бита в регистре сформирует соответствующее PCI-прерывание.

Определение регистров

Описываются четыре регистра состояния и управления контроллера прерывания от PCI и периферийных устройств,

регистр выбора маршрутизации прерывания PCI,

регистр состояния прерывания XINT6,

регистр состояния прерывания XINT7,

регистр состояния прерывания NMI.

Каждый из них является 32-битным регистром и отображен в пространство памяти локального процессора.

Все регистры являются видимыми как регистры, отображенные в память P2P, и к ним можно обратиться через внутреннюю шину памяти. Регистр выбора маршрутизации прерывания PCI доступен с внутренней шины памяти и через пространство регистра конфигурации PCI (функция 0).

Регистр выбора маршрутизации прерывания PCI - PIRSR

Регистр выбора маршрутизации прерывания PCI (PIRSR) определяет маршрутизацию внешних входных линий. Входные линии состоят из 4 входов прерывания PCI, которые направляют или на первичные PCI-прерывания или на прерывания локального процессора. Линии PCI-прерываний определяются как "чувствительные к уровню", установленному низкому уровню. Установка или снятие сигналов на линиях прерываний являются асинхронными к синхронизации PCI или процессора.

Если входы прерывания вторичного PCI направляются на линии прерывания первичного PCI, входы XINT3:0# локального процессора должны быть установлены в неактивное состояние.

Регистр состояния прерывания XINT6 - X6ISR

Регистр состояния прерывания XINT6 (X6ISR) указывает текущие прерывания XINT6 задержки. Источником прерывания XINT6 могут быть внутренние периферийные устройства, подсоединенные через фиксатор прерывания XINT6 или входную линию внешнего XINT6#. Прерывания, которые могут быть сформированы на входе XINT6#, описаны выше применительно к маршрутизации прерывания внутренних периферийных устройств.

X6ISR используется прикладным программным обеспечением для определения источника прерывания на входе XINT6# и для очистки этого прерывания. Все биты этого регистра имеют статус "только для считывания". Биты этого регистра очищаются, когда источники прерывания (источник для регистра состояния, показанного в таблице 11) очищаются. X6ISR будет копировать текущее состояние входа на фиксатор прерывания XINT6.

Из-за асинхронной природы периферийных устройств P2P, могут быть активными множество прерываний, когда прикладное программное обеспечение считывает регистр X6ISR. Прикладное программное обеспечение должно хранить это множество условий прерывания соответствующим образом. К тому же, прикладное программное обеспечение может



впоследствии считать регистр X6ISR для определения, имелись ли дополнительные прерывания во время обработки предыдущих прерываний. Все прерывания от регистра X6ISR должны иметь одинаковый уровень приоритета в локальном процессоре ("Руководство пользователя микропроцессора 1960 Jx" предоставляют описание для установки механизма приоритета прерывания).

Таблица 15 представляет собой определение X6ISR.

Регистр состояния прерывания XINT7 - X7ISR

Регистр состояния прерывания XINT7 (X7ISR) указывает текущие прерывания XINT7 задержки. Источником прерывания XINT7 могут быть внутренние устройства прерывания, подсоединенные через фиксатор прерывания XINT7 или входную линию внешнего XINT7#.

X7ISR используется прикладным программным обеспечением для определения источника прерывания на входе XINT7# и для очистки этого прерывания. Все биты этого регистра имеют статус "только для считывания". Биты этого регистра очищаются, когда источники прерывания (источник для регистра состояния, показанного в таблице 12) очищаются. X7ISR будет копировать текущее состояние входа на фиксатор прерывания XINT7.

Из-за асинхронной природы периферийных устройств P2P могут быть активными множество прерываний, когда прикладное программное обеспечение считывает регистр X7ISR. Прикладное программное обеспечение должно хранить это множество условий прерывания соответствующим образом. К тому же, прикладное программное обеспечение может впоследствии считать регистр X7ISR для определения, имелись ли дополнительные прерывания во время обработки предыдущих прерываний. Все прерывания от регистра X7ISR должны иметь одинаковый уровень приоритета в локальном процессоре,

Таблица 16 представляет собой определение X7ISR.

Регистр состояния прерывания NMI - NISR

Регистр состояния прерывания NMI (NISR) показывает текущие NMI прерывания задержки. Источником NMI прерывания могут быть внутренние периферийные устройства, подсоединенные через фиксатор прерывания NMI или входную линию внешнего NMI#.

Регистр состояния прерывания NMI используется прикладным программным обеспечением для определения источника прерывания на входе NMI# и для очистки этого прерывания. Все биты регистра NISR имеют статус "только для считывания". Биты этого регистра очищаются, когда источники прерывания (источник для регистра состояния, показанного в таблице 13) очищаются. NISR копирует текущее состояние входа на фиксатор прерывания NISR.

Из-за асинхронной природы периферийных устройств P2P могут быть активными множество прерываний, когда прикладное программное обеспечение считывает регистр NISR. Прикладное программное обеспечение должно хранить это множество условий прерывания соответствующим образом. К тому же,

прикладное программное обеспечение может впоследствии считать регистр NISR для определения, имелись ли дополнительные прерывания во время обработки предыдущих прерываний. Все прерывания от регистра NISR должны иметь одинаковый уровень приоритета в локальном процессоре.

Таблица 17 показывает определение битов для считывания регистра состояния прерывания NMI.

Внутренний арбитраж

Ниже следует описание внутреннего арбитража процессора P2P. Он включает в себя арбитраж внутренней локальной шины, также как и арбитраж каждого из PCI-интерфейсов в процессоре. Режим работы, установка, интерфейс внешней памяти и реализация арбитража описана ниже.

Арбитраж локальной шины

Процессор P2P требует механизма арбитража для управления монопольным использованием локальной шины. "Хозяева" шины, подключенные к локальной шине, состоят из трех каналов DMA, устройства преобразования адреса первичной PCI, устройства преобразования адреса вторичной PCI, локального процессора и внешних "хозяев" шины.

Устройство арбитража 57 локальной шины (LBAU) реализует алгоритм равнодоступности, который предоставляет каждому "хозяину" шины возможность захвата управления локальной шиной. Алгоритм объединяет циклическую схему с механизмом приоритета. В предпочтительном варианте воплощения реализация должна позволить прикладной задаче назначать приоритеты каждому ведущему устройству шины независимо.

Устройство арбитража локальной шины ответственно за предоставление локальной шины "хозяину" шины. Все "хозяева" шины содержат логические схемы для удаления самих себя в качестве "хозяина" шины с локальной шины, когда они потеряли свой внутренний сигнал GNT#. Существует программируемый 12-битный счетчик для ограничения времени, в течение которого "хозяин" управляет локальной шиной, и задания времени, когда "хозяин" шины должен освободить право монопольного использования, если другие "хозяева" запрашивают локальную шину.

Ведущие "хозяева" шины могут быть использованы на локальной шине посредством добавления внешней логики для управления HOLD/HOLDA. Процессор P2P позволяет одному внешнему "хозяину" шины участвовать в алгоритме равнодоступности. Если более одного внешнего "хозяина" шины используется на локальной шине, то внешние логические схемы должны отсечь все внешние устройства как одно устройство (определяет HOLD и направляет HOLDA).

Устройство арбитража локальной шины управляет устройством "потери" локального процессора. Это устройство позволяет локальному процессору терять локальную шину для предотвращения возможных тупиковых ситуаций. При "потере" процессор поддерживается в состоянии ожидания (сигнал L\_RDYRCV не активен). Внутренние буферы переводят мультиплексированную шину адресов/данных в третье состояние,



позволяя другим "хозяевам" шины (ATU, DMA и т.д.) управлять шиной и поэтому предотвращать ситуацию, когда выходящая транзакция требует ресурсы, используемые входящей транзакцией. К тому же, устройство "потери" оптимизирует пропускную способность локальной шины во время всей выходящей операции считывания процессором.

В дополнение к устройству арбитража локальной шины процессор P2P содержит два локальных устройства арбитража PCI. Локальное первичное устройство арбитража 55a управляет доступом к внутренней первичной PCI-шине. Арбитраж имеет место для первичной PCI-шины между первичной ATU, DMA каналами 0 и 1 и первичным интерфейсом мостового устройства PCI-PCI. Локальное вторичное устройство арбитража 55b управляет доступом к внутренней вторичной PCI-шине. Арбитраж имеет место для вторичной PCI-шины между вторичной ATU, DMA каналом 2 и вторичным интерфейсом мостового устройства PCI-PCI. Оба локальных устройства арбитража PCI функционируют одинаковым образом. Предпочтительно, логические схемы арбитража должны разрешать управление локальной шиной многими "хозяевами" шины. Когда "хозяин" шины запрашивает локальную шину, то устройство арбитража локальной шины может сначала получить управление локальной шиной от локального процессора посредством установления уровня сигнала запроса HOLD. Локальный процессор может предоставить шину логическим схемам арбитража посредством установления уровня сигнала HOLDA и установлением сигнальных линий процессора в третье состояние. Логические схемы арбитража могут затем предоставить другому "хозяину" локальную шину посредством возвращения соответствующего внутреннего сигнала GNT#.

Арбитраж внутренней PCI-шины

Процессор P2P содержит два внутренних устройства арбитража, которые управляют доступом к внутренним PCI-шинам в устройстве. Фиг.10 изображает схему этих внутренних устройств арбитража и ресурсов, которыми они управляют.

Устройство арбитража первичной внутренней PCI разрешает конфликты для следующих внутренних устройств первичный мостовой интерфейс,

первичное ATU,  
DMA канал 0,  
DMA канал 1.

Вторичное устройство 55b арбитража внутренней PCI разрешает конфликты для следующих внутренних устройств вторичного интерфейса моста, вторичного ATU и DMA канала 2.

Каждое устройство арбитража внутренней PCI использует схему арбитража с фиксированным циклом, в которой каждое устройство на шине имеет одинаковый приоритет.

Схема арбитража с фиксированным циклом интерпретируется следующим образом

после сброса маркер для арбитража принадлежит устройству #1 в каждом внутреннем устройстве PCI-арбитража, арбитраж выполняется по каждому

импульсу, которым устройство устанавливает внутренний сигнал REQ#, направленный к арбитру,

следующий обладатель маркера (т.е. шины) должен иметь ближайший номер устройства к текущему собственнику (или последний, если шина свободна). Например, если устройство #3 является текущим собственником, а устройство #4 и устройство #1 запрашивает шину, то доступ к шине получит устройство #4,

маркер пропускается, когда арбитр активизирует внутренний сигнал предоставления к внутреннему "хозяину" шины. Он является сигналом предоставления от арбитра. Реальные выходные сигналы к "хозяевам" внутренней шины являются еще маскируемыми внешним входным сигналом GNT#.

Теория работы

Каждое устройство на внутренней PCI-шине требует использования PCI-шины для работы в качестве ведущего устройства. Арбитраж имеет место всякий раз, как ресурсы, подключенные к внутренней шине, активизируются сигналом запроса (REQ#). Предоставление осуществляется для следующего ресурса в циклической схеме. Предоставление внутренней PCI-шины связано с состоянием внешней PCI-шины. Состояние внешних линий запроса (P\_REQ# и S\_REQ#) является точной копией логического "ИЛИ" линий запроса на каждой внутренней PCI-шине.

"Хозяин" внутренней PCI-шины может принимать внутренний сигнал GNT# в любое время от внутреннего арбитра PCI (скрытый арбитраж). "Хозяева" внутренней PCI-шины еще являются ответственными за непрерывное управление FRAME#, IRDY# и их внутреннего входного сигнала GNT# для обеспечения того, что монопольное использование шины поддерживается до начала обращения. FRAME# и IRDY# должны иметь высокий уровень, а входной сигнал предоставления должен иметь низкий уровень по спадающему фронту сигнала синхронизации, который определяет период синхронизации, в котором ведущее устройство затем переводит FRAME# в низкий уровень для начала цикла. Устройства арбитража внутренних PCI будут управлять внешними сигналами предоставления (P\_REQ# и S\_REQ#) и будут только подавать внутренний сигнал предоставления, на основании внешних сигналов предоставления, являющихся истинными.

Внутренний PCI-арбитр будет удалять сигнал GNT# "хозяина" внутренней PCI при следующих ситуациях

внешние сигналы предоставления (P\_REQ# и S\_REQ#) становятся неактивными, внутренний арбитр деактивизирует внутренний сигнал предоставления "хозяев" внутренней шины,

текущий обладатель шины снимает свой выходной сигнал REQ#.

Входной сигнал предоставления каждого "хозяина" шины может рассматриваться условием "ИЛИ" внешнего сигнала предоставления и внутреннего сигнала предоставления изнутри устройства арбитража внутренней PCI-шины. Устройство арбитража будет активизировать внутреннее предоставление "хозяину" шины на основании

"победителя" схемы арбитража, но реальный сигнал GNT#, подаваемый на вход ведущего устройства шины, подается от операции "ИЛИ" внутреннего сигнала предоставления с внешним входом предоставления.

Устройство внутреннего арбитража ответственно за обеспечение того, что только один внутренний сигнал GNT# является активным в любой момент времени. Как только "хозяин" внутренней шины теряет внутренний сигнал GNT#, он должен в итоге освободить монопольное использование шиной. Внутренний сигнал GNT# придерживается правил снятия уровня сигнала GNT# в "Спецификации локальной шины PCI" (протокол сигналов арбитража). Другие подробности, необходимые для реализации арбитража внутренней PCI-шины, должны быть легко понятны специалистам в области, к которой относится изобретение.

Интерфейсное устройство шины I<sup>2</sup>C

Ниже следует описание интерфейсного устройства шины I<sup>2</sup>C (шины для межсоединений интегральных схем) процессора P2P. Режимы работы, установка и реализация интерфейсного устройства шины I<sup>2</sup>C описаны ниже.

Обзор

Интерфейсное устройство 61 шины I<sup>2</sup>C позволяет локальному процессору 34 служить в качестве ведущего и подчиненного устройства, постоянно подключенного к шине I<sup>2</sup>C. Шина I<sup>2</sup>C является последовательной шиной, разработанной Philips Corporation и состоящей из двухпроводного интерфейса. SDA является линией данных для входных и выходных функций, а SCL является линией синхронизации для синхронизации и управления шиной I<sup>2</sup>C.

Шина I<sup>2</sup>C позволяет процессору P2P обмениваться с другими периферийными I<sup>2</sup>C устройствами и микроконтроллерами для функций управления системой. Последовательная шина требует минимального аппаратного обеспечения для экономичной системы для передачи информации о надежности и состоянии, находящейся на подсистеме P2P, на внешнее устройство.

Интерфейсное устройство шины I<sup>2</sup>C является периферийным устройством, которое постоянно подключено к внутренней локальной шине P2P. Данные передаются на шину I<sup>2</sup>C и принимаются с нее через буферизованный интерфейс. Информация о состоянии и управлении передается через набор отображенных в память регистров локального процессора. Спецификация шины I<sup>2</sup>C содержит полную информацию о работе I<sup>2</sup>C.

Теория работы

Шина I<sup>2</sup>C определяет законченный последовательный протокол для прохождения информации между исполнительными устройствами на шине, используя только двухпроводный интерфейс. Каждое устройство на шине распознается от уникального 7-битового адреса и может работать в качестве передатчика или приемника. Дополнительно к работе в качестве передатчика или приемника шина I<sup>2</sup>C функционирует в режиме ведущее/подчиненное устройство.

В качестве примера работы шины I<sup>2</sup>C рассмотрим случай микроконтроллера, работающего ведущим устройством на шине. Микроконтроллер в качестве ведущего устройства может адресовать электрически перепрограммируемую память (EEPROM) в качестве подчиненного устройства для приема данных для записи. Микроконтроллер может быть ведущим устройством-передатчиком, а EEPROM - подчиненным устройством-приемником. Если микроконтроллер желает считывать данные, то микроконтроллер должен быть ведущим устройством-приемником, а EEPROM - подчиненным устройством-передатчиком. В обоих случаях ведущее устройство инициирует и прерывает транзакцию.

Шина I<sup>2</sup>C подходит для систем с многими ведущими устройствами, что означает, что более одного устройства могут пытаться инициировать передачу данных в одно и то же время. Шина I<sup>2</sup>C определяет процедуру арбитража для управления такой ситуацией. Если два или более ведущих устройств управляют шиной одновременно, то первое ведущее устройство должно выдавать 1, в то время как другие выдают 0, что приведет к потере арбитража. Это зависит от операции "монтажное И" в линиях SDA и SCL шины I<sup>2</sup>C.

Последовательная работа шины I<sup>2</sup>C использует структуру шины "монтажное И". Это является способом для множества устройств управлять линиями шины и сигнализировать друг другу о событиях, таких как арбитраж, режимы ожидания, условия ошибок и т.д. Например, когда ведущее устройство управляет синхронизацией (SCL) во время передачи данных, оно будет передавать бит в каждом случае, когда сигнал синхронизации имеет высокий уровень. Если подчиненное устройство не может принять или управлять данными со скоростью, которую требует ведущее устройство, оно может удерживать уровень сигнала синхронизации низким между состояниями с высоким уровнем для вставки состояний ожидания. "Монтажное И" реализуется на выходном каскаде устройства. Данные на шине I<sup>2</sup>C могут быть переданы с максимальной скоростью 400 Кбит/сек.

Транзакции I<sup>2</sup>C инициируются или локальным процессором в качестве ведущего устройства или принимаются процессором в качестве подчиненного. Оба случая могут привести к обращениям процессора для считывания, записи или обеих операций к шине I<sup>2</sup>C.

Операционные блоки

Интерфейсное устройство шины I<sup>2</sup>C процессора P2P является подчиненным периферийным устройством, которое подключено к локальной шине. Устройство использует механизм прерывания процессора P2P для извещения локального процессора, что на шине I<sup>2</sup>C имеются активные устройства. Фиг. 11 изображает блок-схему интерфейсного устройства шины I<sup>2</sup>C и его интерфейс с локальной шиной.

Интерфейсное устройство шины I<sup>2</sup>C состоит из двухпроводного интерфейса 61 с шиной I<sup>2</sup>C, 8-битного буфера 61а для передачи данных к локальному процессору и от него, набора регистров управления и

состояния 61b и сдвигового регистра 61c для параллельно-последовательных преобразований.

О прерываниях I<sup>2</sup>C оповещаются через прерывание XINT7# процессора и регистр состояния прерывания XINT7 (X7ISR).

Интерфейсное устройство шины I<sup>2</sup>C устанавливает бит в регистре X7ISR, когда буфер заполнен, определен адрес подчиненного устройства, потерян арбитраж или имеет место условие ошибки шины. Все условия прерывания очищаются явно локальным процессором.

Регистр буфера данных I<sup>2</sup>C является 8-битным регистром данных, который принимает байтовые данные от интерфейса регистра сдвига на шину I<sup>2</sup>C на одной стороне и параллельные данные от локальной шины процессора P2P с другой стороны. Последовательный регистр сдвига недоступен для пользователя.

Интерфейсное устройство шины APIC

Ниже следует описание интерфейсного устройства 63 шины APIC, которое обеспечивает механизм обмена между локальной шиной и трехпроводной шиной APIC. Оно обеспечивает две основные функции:

оно дает локальному процессору возможность посылать сообщение о прерывании на шину APIC и, необязательно, быть прервано, когда сообщение послано. Локальный процессор может затем считывать результирующее состояние передачи сообщения для проверки появления ошибок,

оно также может принимать сообщение EOI от шины APIC и, необязательно, прерывать локальный процессор для информирования его, что вектор EOI доступен.

Режимы работы, установка и реализация интерфейса описаны ниже.

Обзор архитектуры APIC

Архитектура интерфейса APIC определяется как архитектура прерывания для всех систем, совместимых с микропроцессорной спецификацией (MPS). MPS, версия 1.1 доступна от Intel Corporation, Order N 242016-003. Основными особенностями архитектуры APIC являются следующие

1. APIC предоставляет управление прерываниями микропроцессора с Intel-архитектурой, такого как процессоры Pentium на 90 и 100 МГц, обеспечивая статическое и динамическое симметричное распределение прерываний среди всех процессоров.

2. Динамическое распределение прерываний включает в себя маршрутизацию прерывания к процессору с наимизшим приоритетом.

3. APIC работает в системах с множеством подсистем ввода/вывода, в которых каждая подсистема может иметь свой собственный набор прерываний.

4. APIC обеспечивает межпроцессорные прерывания, позволяя любому процессору прерывать любой процессор или набор процессоров, включая самого себя.

5. Каждая входная линия прерывания APIC является индивидуально программируемой программным обеспечением как срабатывающей от фронта или от уровня сигнала. Вектор прерывания и

информация управления прерываниями может быть указана для каждой линии.

6. APIC поддерживает именную/адресную схему, которая может быть урезана программным обеспечением для "включения" множества архитектур систем и моделей использования.

7. APIC поддерживает общесистемные функции управления процессором, относящиеся к NMI, INIT и прерыванию управления системой (SMI).

8. APIC совместно существует с 8259A PIC для поддержания совместимости PC.

9. APIC предоставляет программируемый приоритет прерываний (вектора) для каждой входной линии прерывания. Так как интерфейс программирования APIC состоит из двух местоположений 32-битовых памяти, функциональные особенности APIC ввода/вывода могут быть эмулированы локальным процессором в процессоре P2P.

Специфические подробности реализации для APIC ввода/вывода, подходящие для использования в изобретении, могут быть ясны специалистам в области, к которой относится изобретение.

### Формула изобретения:

1. Интегральная схема соединения первой внешней шины межсоединения периферийных компонент со второй внешней шиной межсоединения периферийных компонент, содержащая первую внутреннюю шину для подсоединения к указанной первой внешней шине, вторую внутреннюю шину для подсоединения к указанной второй внешней шине, шинный мост, имеющий первый шинный интерфейс, подсоединенный к указанной первой внутренней шине, и второй шинный интерфейс, подсоединенный к указанной второй внутренней шине, причем указанный шинный мост позволяет пропускать транзакции между первой и второй внешними шинами, локальный процессор, имеющий пространство для локального адреса, подсоединенный к указанному шинному мосту для обработки транзакций, принятых от внешних источников, и транзакций, поступающих на указанный первый шинный интерфейс и указанный второй шинный интерфейс, локальную шину для прохождения данных между указанным процессором и памятью, доступ к которой осуществляется в указанном пространстве локального адреса, контроллер памяти для управления записями и считываниями из указанной памяти с использованием данных и адресов на указанной локальной шине.

2. Интегральная схема по п.1, отличающаяся тем, что она дополнительно содержит первую схему преобразования адреса, подсоединенную к указанной локальной шине и указанной первой внутренней шине, предназначенную для преобразования адресов входящих транзакций на указанной первой внутренней шине и адреса, используемые указанным локальным процессором, и для преобразования адресов выходящих транзакций на указанной локальной шине в адреса, используемые устройствами, подсоединенными к указанной первой внешней шине, вторую схему преобразования адреса, подсоединенную к указанной локальной шине и указанной второй внутренней шине, предназначенную для

преобразования адресов входящих транзакций на указанную вторую внутреннюю шину в адреса, используемые указанным локальным процессором, и для преобразования адресов выходящих транзакций на указанной локальной шине в адреса, используемые устройствами, подключенными к указанной второй внешней шине.

3. Интегральная схема по п.2, отличающаяся тем, что она дополнительно содержит схему передачи сообщений, подсоединенную к указанной первой схеме преобразования адреса и предназначенную для формирования прерывания для использования указанным локальным процессором, когда новые данные поступают на указанную первую внутреннюю шину, и для формирования прерывания, по меньшей мере, на одной линии прерывания указанной первой внутренней шины, когда указанный локальный процессор помещает данные на указанную локальную шину для использования устройствами, подсоединенными к указанной внешней шине.

4. Интегральная схема по п.2, отличающаяся тем, что она дополнительно содержит первое средство арбитража шины для управления доступом к указанной первой внутренней шине между указанным первым шинным интерфейсом и указанной первой схемой преобразования адреса, второе средство арбитража шины для управления доступом к указанной второй внутренней шине между указанным вторым шинным интерфейсом и указанной второй схемой преобразования адреса.

5. Интегральная схема по п.2, отличающаяся тем, что она дополнительно содержит средство арбитража локальной шины для управления доступом к указанной локальной шине от указанного локального процессора, указанной первой схемы преобразования адреса и указанной второй схемы преобразования адреса.

6. Интегральная схема по п.1, отличающаяся тем, что она дополнительно содержит первый контроллер прямого доступа к памяти, подсоединенный к указанной локальной шине и указанной первой внутренней шине для передачи блоков данных между указанной первой внутренней шиной и указанной локальной памятью, второй контроллер прямого доступа к памяти, подсоединенный к указанной локальной шине и указанной второй внутренней шине для передачи блоков данных между указанной второй внутренней шиной и указанной локальной памятью.

7. Интегральная схема по п.1, отличающаяся тем, что она дополнительно содержит контроллер прерываний периферийных устройств для маршрутизации прерываний от множества источников прерываний на вход единственного прерывания указанного локального процессора, регистр состояния, определяющий, какой из указанного множества источников прерывания вызвал указанное единственное прерывание.

8. Интегральная схема по п.1, отличающаяся тем, что она дополнительно содержит шину межсоединения интегральных схем для подсоединения к внешним датчикам, предназначенную для управления системой с

использованием указанной интегральной схемы, интерфейс межсоединения интегральных схем для разрешения указанному локальному процессору служить в качестве ведущего и подчиненного устройства, постоянно находящегося на указанной шине.

9. Интегральная схема по п.1, отличающаяся тем, что она дополнительно содержит шину усовершенствованного программируемого контроллера прерываний для подключения к внешним процессорам, интерфейс шины для разрешения обмена между указанными внешними процессорами и указанным локальным процессором.

10. Интегральная схема по п.1, отличающаяся тем, что указанный шинный мост дополнительно содержит буфер пересылки вниз, включенный между указанным первым шинным интерфейсом и указанным вторым шинным интерфейсом, буфер пересылки вверх, включенный между указанным первым шинным интерфейсом и указанным вторым шинным интерфейсом, набор регистров конфигурации, подключенный к указанному первому шинному интерфейсу.

11. Интегральная схема по п.1, отличающаяся тем, что указанный локальный процессор содержит микропроцессор.

12. Процессорная система, включающая в себя, по меньшей мере, один главный процессор, подсоединенный к первой шине межсоединения периферийных компонент, и, по меньшей мере, одно периферийное устройство, подсоединенное к второй шине, содержащая в единственной интегральной схеме первую внутреннюю шину, подсоединенную к указанной первой шине, вторую внутреннюю шину, подсоединенную к указанной второй шине, шинный мост, имеющий первый шинный интерфейс, подсоединенный к указанной первой внутренней шине, и второй шинный интерфейс, подсоединенный к указанной второй внутренней шине, причем указанный шинный мост позволяет пропускать транзакции между указанной первой шиной и указанной второй шиной, локальный процессор, подсоединенный к указанному шинному мосту для обработки транзакций, принятых от внешних источников, и транзакций, поступающих на указанный первый шинный интерфейс и указанный второй шинный интерфейс, локальную шину для прохождения данных между указанным процессором и памятью, контроллер памяти для управления записями и считываниями из указанной внешней памяти с использованием данных и адресов на указанной локальной шине.

13. Система по п.12, отличающаяся тем, что указанная интегральная схема дополнительно содержит первую схему преобразования адреса, подсоединенную к указанной локальной шине и указанной первой внутренней шине, предназначенную для преобразования адресов входящих транзакций на указанной первой внутренней шине в адреса, используемые указанным локальным процессором, и для преобразования адресов выходящих транзакций на указанной локальной шине в адреса, используемые устройствами, подключенными к указанной первой шине

межсоединения первичных компонент, вторую схему преобразования адреса, подсоединенную к указанной локальной шине и указанной второй внутренней шине, предназначенную для преобразования адресов входящих транзакций на указанную вторую внутреннюю шину в адреса, используемые указанным средством локального процессора, и для преобразования адресов выходящих транзакций на указанной локальной шине в адреса, используемые устройством, подсоединенным к указанной второй шине.

14. Система по п.13, отличающаяся тем, что указанная интегральная схема дополнительно содержит схему передачи сообщений, подсоединенную к указанной первой схеме преобразования адреса, предназначенную для формирования прерывания для использования указанным локальным процессором, когда новые данные поступают на указанную первую внутреннюю шину, и для формирования прерывания, по меньшей мере, на одной линии прерывания указанной первой внутренней шины, когда указанный локальный процессор помещает данные на указанную локальную шину для использования устройством, подсоединенным к указанной первой шине.

15. Система по п.13, отличающаяся тем, что указанная интегральная схема дополнительно содержит первое средство арбитража шины для управления доступами к указанной первой внутренней шине между указанным первым интерфейсом шины и указанной первой схемой преобразования адреса, второе средство арбитража шины для управления доступами к указанной второй внутренней шине между указанным вторым интерфейсом шины и указанной второй схемой преобразования адреса.

16. Система по п.13, отличающаяся тем, что указанная интегральная схема дополнительно содержит средство арбитража локальной шины для управления доступом к указанной локальной шине от указанного локального процессора, указанной первой схемой преобразования адреса и указанной второй схемы преобразования адреса.

17. Система по п.12, отличающаяся тем, что указанная интегральная схема дополнительно содержит первый контроллер прямого доступа к памяти, подсоединенный к указанной локальной шине для передачи блоков данных между указанной первой внутренней шиной и указанной локальной памятью, второй контроллер прямого доступа к памяти, подсоединенный к указанной локальной шине и указанной второй внутренней шине для передачи блоков данных между указанной второй внутренней шиной и указанной локальной памятью.

18. Система по п.17, отличающаяся тем, что указанная интегральная схема дополнительно содержит первое средство арбитража шины для управления доступами к указанной первой внутренней шине от

указанного первого интерфейса шины, указанной первой схемы преобразования адреса и указанного первого контроллера, второе средство арбитража шины для управления доступами к указанной второй внутренней шине от указанного второго интерфейса шины, указанной второй схемы преобразования адреса и указанного второго контроллера.

19. Система по п.12, отличающаяся тем, что указанная интегральная схема дополнительно содержит контроллер прерываний периферийных устройств для маршрутизации прерываний от множества источников прерываний на вход единственного прерывания указанного локального процессора, регистр состояния, определяющий, какой из указанного множества источников прерывания вызвал указанное единственное прерывание.

20. Система по п.12, отличающаяся тем, что указанная интегральная схема дополнительно содержит шину межсоединения интегральных схем для подключения к внешним датчикам, предназначенным для управления системой с использованием указанной интегральной схемы, интерфейс для разрешения указанному локальному процессору служить в качестве ведущего и подчиненного устройства, находящегося на указанной шине.

21. Система по п.12, отличающаяся тем, что указанная интегральная схема дополнительно содержит шину усовершенствования программируемого контроллера прерываний для подключения к внешним процессорам, интерфейс для разрешения обмена между указанными внешними процессорами и указанным локальным процессором.

22. Система по п.12, отличающаяся тем, что указанный шинный мост дополнительно содержит буфер пересылки вниз, включенный между указанным первым шинным интерфейсом и указанным вторым шинным интерфейсом, буфер пересылки вверх, включенный между указанным первым шинным интерфейсом и указанным вторым шинным интерфейсом, набор регистров конфигурации, подключенный к указанному первому шинному интерфейсу.

23. Система по п.12, отличающаяся тем, что локальный процессор содержит микропроцессор.

24. Система по п.12, отличающаяся тем, что она дополнительно содержит первое средство арбитража шины для управления доступами к указанной первой внутренней шине от указанного первого интерфейса шины, указанной первой схемы преобразования адреса и указанного первого контроллера прямого доступа к памяти, второе средство арбитража шины для управления доступами к указанной второй внутренней шине от указанного второго интерфейса шины, указанной второй схемы преобразования адреса и указанного второго контроллера.

Таблица 1

## PCI команды

C/BE#	PCI команда	Инициатор:	Инициатор:
		первичная шина Адресат: вторичная шина	первичная шина Адресат: первичная шина
0000 <sub>2</sub>	подтверждение прерывания	игнорируется	игнорируется
0001 <sub>2</sub>	специальный цикл	игнорируется	игнорируется
0010 <sub>2</sub>	считывание ввода/вывода	передается	передается
0011 <sub>2</sub>	запись ввода/вывода	передается	передается
0100 <sub>2</sub>	зарезервирована	игнорируется	игнорируется
0101 <sub>2</sub>	зарезервирована	игнорируется	игнорируется
0110 <sub>2</sub>	считывание из памяти	передается	передается
0111 <sub>2</sub>	запись в память	передается	передается
1000 <sub>2</sub>	зарезервирована	игнорируется	игнорируется
1001 <sub>2</sub>	зарезервирована	игнорируется	игнорируется
1010 <sub>2</sub>	считывание конфигурации	передается	передается
1011 <sub>2</sub>	запись конфигурации	передается	передается
1100 <sub>2</sub>	коэффициент считывания из памяти	передается	передается
1101 <sub>2</sub>	двойной адресный цикл	игнорируется	игнорируется
1110 <sub>2</sub>	шина считывания памяти	передается	передается
1111 <sub>2</sub>	запись в память и недостовверные данные	передается	передается

RU 2157000 C2

RU 2157000 C2

Таблица 2

Прохождение транзакций	Пропустить принятую пересланную "запись в память"?	Пропустить принятую задержанную "запрос на считывание"?	Пропустить принятую задержанную "запрос на запись"?	Пропустить принятую задержанную "завершение считывания"?	Пропустить принятую задержанную "завершение записи"?
Новая	Нет	Да	Да	Да	Да
пересланная "запись в память"					
Новая задержанная "запрос на считывание"	Нет	Не принимать	Не принимать	Нет	Да
Новая задержанная "запрос на запись"	Нет	Не принимать	Не принимать	Нет	Да
Новая задержанная "завершение считывания"	Нет	Да	Да	Не принимать	Не принимать
Новая задержанная "завершение записи"	Да	Да	Да	Нет	Не принимать

RU 2157000 C2

RU 2157000 C2

Таблица 3

Адресация регистров конфигурации моста PCI-PCI

Имя регистра	Размер в байтах	Смещение адреса
Регистр ID продавца - VIDR	2	00H
Регистр ID устройства - DIDR	2	02H
Первичный регистр команд - PCMDR	2	04H
Первичный регистр состояния - PSR	2	06H
Регистр ID версии - RIDR	1	08H
Регистр кода класса - CCR	3	09H
Регистр размера кэша - CLSR	1	0CH
Первичный регистр таймера задержки - PLTR	1	0DH
Регистр типа заголовка - HTR	1	0EH
Регистр номера первичной шины - PBNR	1	18H
Регистр номера вторичной шины - SBNR	1	19H
Регистр номера подчиненной шины - SubBNR	1	1AH
Вторичный регистр таймера задержки - SLTR	1	1BH
Базовый регистр ввода/вывода - IOBR	1	1CH
Регистр границы ввода/вывода - IOLR	1	1DH
Вторичный регистр состояния - SSR	2	1EH
Базовый регистр памяти - MBR	2	20H
Регистр границы памяти - MLR	2	22H

RU 2157000 C2

RU 2157000 C2



Базовый регистр памяти с предварительной выборкой - PMBR	2	24H
Регистр границы памяти с предварительной выборкой - PMLR	2	26H
Регистр управления мостом - BCR	2	3EH
Регистр управления расширенными функциями моста - EBCR	2	40H
Вторичный регистр выбора IDSEL-SISR	2	42H
Первичный регистр состояния прерывания моста - PBISR	4	44H
Вторичный регистр состояния прерывания моста - SBISR	4	48H
Вторичный регистр управления арбитражем - SACR	4	4CH
Регистр выбора маршрутизации прерывания PCI-PIRSR	4	50H
Вторичный базовый регистр ввода/вывода - SIOBR	1	54H
Вторичный регистр границы ввода/вывода - SIOLR	1	55H
Вторичный базовый регистр памяти - SMBR	2	58H
Вторичный регистр границы памяти - SMLR	2	5AH
Вторичный регистр декодирования - SDER	2	5CH

Таблица 4а

Регистр управления расширенными функциями моста - EBCR

	По умолчанию	Считывание/ запись	Описание
15:07	000000000 <sub>2</sub>	Только считывание	Зарезервирован
06	Изменяется с изменением внешнего состояния линии CONFIG_MODE при сбросе первичной PCI шины	Считывание/ запись	Цикл конфигурации блоки- рован - когда бит установлен, первичный PCI интерфейс процессора P2P будет реагировать на все циклы конфигурации с условием Retry. Когда бит сброшен, процессор P2P будет реагировать на соответствующие циклы

RU 2157000 C2

RU 2157000 C2

			<p>конфигурации. Значение по умолчанию для этого бита основано на внешнем состоянии линии CONFIG_MODE при нарастающем фронте сигнала P_RST#.</p> <p>Если внешнее состояние линии соответствует высокому уровню сигнала, то бит является установленным. Если внешнее состояние линии соответствует низкому уровню, то бит очищен.</p>
05	0 <sub>2</sub>	Только считывание	Зарезервирован
04	0 <sub>2</sub>	Только считывание	Зарезервирован
03	Изменяется в зависимости от внешнего состояния линии режима SYNC# при сбросе	Только считывание	<p>Режим SYNC#-описывает, какой из трех сигналов синхронизации является синхронным: первичной PCI шины, вторичной PCI шины и локального процессора. Если бит очищен, все три сигнала си-</p>

	первичной PCI шины		нхронизации являются синхронными. Если бит установлен, то синхронизация первичной PCI шины является асинхронной по отношению к синхронизации вторичной PCI шины и синхронизации локального процессора. Значения по умолчанию для этого бита основаны на внешнем состоянии линии SYNC# при нарастающем фронте P_RST#.
02	0 <sub>2</sub>	Считывание/ запись	Сброс моста - когда бит установлен, то весь мост PCI-PCI должен быть установлен в исходное состояние (сброшен). Все регистры моста должны быть установлены в свои значения по умолчанию (за исключением бита сброса вторичной шины регистра BCR), все конечные автоматы должны быть сброшены, а все бу-

			<p>феры очищены. Бит сброса вторичной шины в регистре BCR должен быть установлен за гарантированное минимальное время сброса PCI. Программное обеспечение должно быть запрошено на сброс этого бита для снятия сигнала сброса вторичной шины.</p>
01	<p>Изменяется в зависимости от внешнего состояния линии RST_MODE при сбросе первичной PCI шины</p>	<p>Считывание/запись</p>	<p>Сброс процессора - этот бит устанавливает в начальное состояние только локальный процессор без сброса вторичной стороны моста. Установка этого бита переводит процессор в начальное состояние и поддерживает его. Программное обеспечение должно быть запрошено для очистки этого бита для снятия сигнала сброса локального процессора. Значение по умолчанию для этого бита основано на внешнем состоя-</p>

			нии линии RST_MODE при нарастающем фронте P_RST#. Если внешнее состояние линии соответствует высокому уровню сигнала, то бит является установленным. Если внешнее состояние соответствует низкому уровню сигнала, то бит сброшен.
00	C <sub>2</sub>	Считывание/ запись	Блокировка пересылки - если этот бит установлен, то мосту не разрешается пересылать транзакции записи от любого интерфейса моста. Все транзакции обрабатываются как задержанные транзакции. Если этот бит очищен, то мосту разрешается пересылать транзакции записи.

Таблица 4b

Регистр состояния прерывания первичного моста - PBISR

Бит	По умолчанию	Считывание/ запись	Описание
31:05	0000000H	Только считывание	Зарезервирован
04	02	Считывание/ очистка	Уровень сигнала P_SERR# установлен - этот бит установлен, если на пер- вичной PCI шине установ- лен уровень сигнала P_SERR#
03	02	Считывание/ очистка	Преждевременное прерыва- ние работы ведущего уст-

RU 2157000 C2

RU 2157000 C2



			ройства PCI - этот бит является установленным всякий раз когда транзакция, инициированная первичным ведущим интерфейсом, заканчивается в преждевременно прерванном ведущем устройстве.
02	02	Считывание/ очистка	Прекращение работы устройства-адресата PCI (ведущего) - этот бит является установленным всякий раз когда транзакция, инициированная первичным ведущим интерфейсом, преждевременно прерванном ведущем устройстве.
01	02	Считывание/ очистка	Сброс устройства-адресата PCI (адресата)- этот бит является установленным всякий раз когда первичный интерфейс, выступающий как адресат, прерывает транзакцию на PCI шине с преждевременным прерыванием адре-

			сата.
00	02	Считывание/ очистка	<p>Ошибка четности ведущей PCI - первичный интерфейс устанавливает этот бит, когда удовлетворяются три условия:</p> <p>1)исполнительное устройство шины устанавливает уровень сигнала P_PERR# самостоятельно или обнаруживает установленный уровень сигнала P_PERR#,</p> <p>2)исполнительное устройство, устанавливающее бит, действует как "хозяин" шины для операции, в которой имеет место ошибка,</p> <p>3)бит отклика на ошибку четности (регистр команд) установлен.</p>

Таблица 4с

Регистр состояния прерывания вторичного моста - SBISR

Бит	По умолчанию	Считывание/ запись	Описание
31:05	0000000H	Только считывание	Зарезервирован
04	02	Считывание/ очистка	Уровень сигнала P_SERR# установлен - этот бит ус- тановлен, если на вторич- ной PCI шине установлен уровень сигнала P_SERR#
03	02	Считывание/ очистка	Преждевременное прерыва- ние работы ведущего PCI - этот бит является уста- новленным всякий раз, когда транзакция, инициа-

RU 2157000 C2

RU 2157000 C2

			рованная вторичным ведущим интерфейсом, заканчивается в преждевременно прерванном ведущем устройстве.
02	02	Считывание/ очистка	Прекращение работы устройства-адресата PCI (ведущего) - этот бит является установленным всякий раз когда транзакция, инициированная вторичным ведущим интерфейсом, преждевременно прерванном ведущем устройстве.
01	02	Считывание/ очистка	Сброс устройства-адресата PCI (адресата) - этот бит является установленным всякий раз когда вторичный интерфейс, выступающий как адресат, прерывает транзакцию на PCI шине с преждевременным прерыванием адресата.
00	02	Считывание/ очистка	Ошибка четности ведущей PCI- вторичный интерфейс устанавливает этот бит, когда удовлетворяются три

			<p>условия:</p> <p>1)исполнительное устройс- тво шины устанавливает уровень сигнала P_PERR# самостоятельно или обна- руживает установленный уровень сигнала P_PERR#,</p> <p>2)исполнительное устройс- тво, устанавливающее бит, действует как "хозяин" шины для операции, в которой имеет место ошибка,</p> <p>3)бит отклика на ошибку четности (регистр команд) установлен.</p>
--	--	--	--

## Вторичный регистр выбора IDSEL

Бит	По умолчанию	Считывание/ запись	Описание
04	0 <sub>2</sub>	Считывание/ запись	AD20-IDSEL блокирован- Когда этот бит установлен, AD20 должен быть снят для любого возможного преобра- зования типа 1 к типу 0. Если бит очищен, то уровень сигнала AD20 должен быть установлен, когда первичные адреса AD[15:11]=00100 <sub>2</sub> во время преобразования типа 1 к типу 0.
03	0 <sub>2</sub>	Считывание/ запись	AD19-IDSEL блокирован- Когда этот бит установлен,

RU 2157000 C2

RU 2157000 C2

			AD19 должен быть снят для любого возможного преоб- разования типа 1 к типу 0. Если бит очищен, то уровень сигнала AD19 должен быть установлен, когда первичные адреса $AD[15:11]=00011_2$ во время преобразования типа 1 к типу 0.
02	$0_2$	Считывание/ запись	AD18-IDSEL блокирован- Когда этот бит установлен, AD18 должен быть снят для любого возможного преоб- разования типа 1 к типу 0. Если бит очищен, то уровень сигнала AD18 должен быть установлен, когда первичные адреса $AD[15:11]=00010_2$ во время преобразования типа 1 к типу 0.
01	$0_2$	Считывание/ запись	AD17-IDSEL блокирован- Когда этот бит установлен, AD17 должен быть снят для любого возможного преоб- разования типа 1 к типу 0. Если бит очищен, то уровень сигнала AD17 должен быть



			установлен, когда первичные адреса $AD[15:11]=00001_2$ во время преобразования типа 1 к типу 0.
00	$0_2$	Считывание/ запись	AD16-IDSEL      блокирован- Когда этот бит установлен, AD16 должен быть снят для любого возможного преобразования типа 1 к типу 0. Если бит очищен, то уровень сигнала AD16 должен быть установлен, когда первичные адреса $AD[15:11]=00000_2$ во время преобразования типа 1 к типу 0.

Таблица 4е

Программируемое управление приоритетом

2-битовое запрограммированное значение	Уровень приоритета
$00_2$	Высокий приоритет
$01_2$	Средний приоритет
$10_2$	Низкий приоритет
$11_2$	Блокировка

## Вторичный базовый регистр ввода/вывода - SIOBR

Бит	По умолчанию	Считывание/ запись	Описание
07:04	0H	Считывание/ запись	Вторичный базовый адрес ввода/вывода- Это поле программируется S_AD[15:12] нижней части положительно декодирован- ного вторичного адресного диапазона ввода/вывода, который должен быть пропущен от вторичной к первичной стороне моста.
03:00	0H	Только считывание	Характеристика адресации ввода/вывода- Значение 0H означает, что мост под- держивает только 16-бит- ную адресацию.

RU 2157000 C2

RU 2157000 C2

## Вторичный регистр границы ввода/вывода - SIOLR

Бит	По умолчанию	Считывание/ запись	Описание
07:04	0H	Считывание/ запись	Вторичный адрес границы ввода/вывода- Это поле программируется  S_AD[15:12] верхнего положительно декодированного адресного диапазона  ввода/вывода, который должен быть пропущен от вторичного к первичному интерфейсу.
03:00	0H	Только	Характеристика вторичной адресации ввода/вывода -  Значение 0H означает, что мост поддерживает только 16-битную адресацию.

RU 2157000 C2

RU 2157000 C2

## Вторичный регистр базовой памяти - SMBR

Бит	По умолчанию	Считывание/ запись	Описание
15:04	000H	Считывание/ запись	Вторичный базовый адрес памяти -Это поле программируется S_AD[31:20] нижней части положительно декодированного адресного диапазона вторичной памяти, который должен быть пропущен от вторичного к первичному интерфейсу.
03:00	0H	Только считывание	Зарезервированы

RU 2157000 C2

RU 2157000 C2

Таблица 4i

## Вторичный регистр границы памяти - SMLR

Бит	По умолчанию	Считывание/ запись	Описание
15:04	000H	Считывание/ запись	Вторичный адрес границы памяти - Это поле программируется S_AD[31:20] верхних значений вторичного адресного диапазона, который должен быть пропущен от вторичной к первичной стороне.
03:00	0H	Только считывание	Зарезервирован

RU 2157000 C2

RU 2157000 C2

## Вторичный регистр разрешения декодирования - SDER

Бит	По умолчанию	Считывание/ запись	Описание
15:04	000000000000 <sub>2</sub>	Только считывание	Зарезервирован
03	0 <sub>2</sub>	Считывание/ запись	Разрешение приватного адресного пространства - когда бит установлен, он блокирует передачу мостом адресов в адресном диапазоне SMBR/SMLR. Это создает приватное адресное пространство на вторичной PCI шине, что разблокирует транзакции "равный к равному".
02	0 <sub>2</sub>	Считывание/ запись	Разрешение вторичного декодирования с вычитанием - Когда этот бит установлен, он разрешает вторичному интерфейсу использовать декодирование с вычитанием (5 импульсов после установления уровня S_FRAME#) для востребования транзакций на

RU 2157000 C2

RU 2157000 C2

			вторичной шине. Любая тран- закция, не востребованная 4-мя импульсами после S_FRAME#, будет востребова- на вторичным интерфейсом на 5-м импульсе и передана в первичный PCI интерфейс.
01	0 <sub>2</sub>	Считывание/ запись	Разрешение вторичного поло- жительного декодирования памяти - когда бит установ- лен, он разрешает вторично- му интерфейсу положительно декодировать адреса памяти на вторичной шине. Адреса в адресном диапазоне SMBR/SMLR должны быть пере- даны через мост. Обратное декодирование должно быть заблокировано.
00	0 <sub>2</sub>	Считывание/ запись	Разрешение вторичного поло- жительного декодирования ввода/вывода - когда бит установлен, он разрешает вторичному интерфейсу мос- тового устройства положи- тельно декодировать адреса ввода/вывода на вторичной



			шине. Адреса в адресной паре SIOBR/SIOLR должны быть переданы через мост. Обратное декодирование должно быть заблокировано.
--	--	--	---

Таблица 5

Поддержка команд ATU

Тип PCI-команды	Востребованные на выход транзакции	Сформированная выходящими транзакциями
Считывание ввода/вывода	Нет	Да
Запись ввода/вывода	Нет	Да
Считывание памяти	Да	Да
Запись в память	Да	Да
Запись в память и недостоверные данные	Да	Нет
Шина считывания памяти	Да	Нет
Коэффициент считывания из памяти	Да	Нет
Считывание конфигурации	Да	Да
Запись конфигурации	Да	Да
Двойной адресный цикл	Нет	Да

RU 2157000 C2

RU 2157000 C2

Таблица 6

Регистры PCI-конфигурации устройства преобразования адресов

Имя регистра	Размер регистра в битах	Номер регистра цикла PCI-конфигурации	Смещение адреса цикла локального процессора
Идентификатор производителя ATU	16	0	00H
Идентификатор устройства ATU	16	0	02H
Регистр команд первичного ATU	16	1	04H
Регистр состояния первичного ATU	16	1	06H
Идентификатор версии ATU	8	2	08H
Код класса ATU	24	2	09H
Размер кэша ATU	8	3	0CH
Таймер задержки ATU	8	3	0DH
Тип заголовка ATU	8	3	0EH
Регистр BIST	8	3	0FH
Базовый адрес первичного входного ATU	32	4	10H
Зарезервирован	32	5	14H
Зарезервирован	32	6	18H

RU 2157000 C2

RU 2157000 C2

Зарезервирован	32	7	1CH
Зарезервирован	32	8	20H
Зарезервирован	32	9	24H
Зарезервирован	32	10	28H
Зарезервирован	32	11	2CH
Базовый адрес расширенного ROM	32	12	30H
Зарезервирован	32	13	34H
Зарезервирован	32	14	38H
Шина прерывания ATU	8	15	3CH
Линия прерывания ATU	8	15	3DH
Минимальное предоставление ATU	8	15	3EH
Максимальная задержка ATU	8	15	3FH
Регистр границы первичного входного ATU	32	16	40H
Регистр значения преобразования первичного входного ATU	32	17	44H
Регистр базового адреса вторичного входного ATU	32	18	48H
Регистр границы вторичного входного ATU	32	19	4CH
Регистр значения преобразования вторичного входного ATU	32	20	50H
Первичный выходной	32	21	54H

регистр значения окна памяти			
Зарезервирован	32	22	58H
Первичный выходной регистр окна ввода/вывода	32	23	5CH
Первичный выходной регистр значения окна DAC	32	24	60H
Первичный выходной регистр верхних битов 64-битного DAC	32	25	64H
Вторичный выходной регистр значения окна памяти	32	26	68H
Вторичный выходной регистр окна ввода/вывода	32	27	6CH
Зарезервирован	32	28	70H
Регистр границы расширенной памяти ROM	32	29	74H
Регистр значения преобразования расширенной памяти ROM	32	30	78H
Зарезервирован	32	31	7CH
Зарезервирован	32	32	80H
Зарезервирован	32	33	84H
Регистр конфигурации ATU	32	34	88H
Зарезервирован	32	35	8CH
Регистр состояния преры-	32	36	90H

вания первичного ATU			
Регистр состояния прерывания вторичного ATU	32	37	94H
Регистр команд вторичного ATU	32	38	98H
Вторичный выходной регистр значения окна DAC	32	39	9CH
Вторичный выходной регистр верхних битов 64-битного DAC	32	40	A0H
Первичный выходной адресный регистр цикла конфигурации	32	41	A4H
Первичный выходной адресный регистр цикла конфигурации	32	42	A8H
Первичный выходной регистр данных цикла конфигурации	32	Не доступен в пространстве конфигурации PCI	ACH
Вторичный выходной регистр данных цикла конфигурации	32	Не доступен в пространстве конфигурации PCI	B0H
Зарезервирован	32	45	B4H
Зарезервирован	32	46	B8H
Зарезервирован	32		BCH
Зарезервирован	32		C0H

Таблица 7а

Регистр границы первичного входного АТУ - PILR

Бит	По умолчанию	Считывание/ запись	Описание
31:04	000000H	Только считывание	Первичная входная граница преобразования - Это значение считывания только что записанной информации, которое определяет размер блока памяти, требуемого для устройства преобразования первичного АТУ
03:00	0000 <sub>2</sub>	Только считывание	Зарезервирован

RU 2157000 C2

RU 2157000 C2

## Регистр значения преобразования первичного входного

ATU - PIATVR

Бит	По умолчанию	Считывание/ запись	Описание
31:02	0000.0800H	Только считывание	Значение преобразования первичного входного ATU - Этот регистр содержит зна- чение для преобразования адреса первичной PCI-шины в локальные адреса. Значение преобразования первичного входящего адреса должно быть словом, выровненным на локальной шине. Адрес по умолчанию разрешает обраше- ние устройства преобразова- ния к внутренним отображен- ным в память регистрам ло- кального процессора P2P
01:00	00 <sub>2</sub>	Только считывание	Зарезервирован

RU 2157000 C2

RU 2157000 C2



Таблица 7с

Регистр базового адреса вторичного входящего ATU - SIABAR

Бит	По умолчанию	Считывание/ запись	Описание
31:04	XXXXXXXXH	Считывание/ запись	Вторичный базовый адрес преобразования- эти биты определяют реальное положение, которому должна соответствовать функция вторичного преобразования, когда она адресуется из вторичной PCI-шины. Размер блока по умолчанию является неопределенным.
03	1 <sub>2</sub>	Только	Индикатор предварительной вы-

RU 2157000 C2

RU 2157000 C2

		считывание	Борки - этот бит определяет области памяти, используемые как память с предварительной выборкой.
02:01	00 <sub>2</sub>	Только считывание	Тип адреса - Эти биты определяют, где может быть расположен блок памяти. Базовый адрес должен быть расположен в любом месте в первых 4 гигабайтах адресного пространства (младшие 32 бита адреса).
00	0 <sub>2</sub>	Только считывание	Индикатор области памяти - Это битовое поле указывает, что содержимое регистра описывает базовый адрес пространства памяти или ввода/вывода. АТУ не занимает область ввода/вывода, таким образом, этот бит должен быть равен нулю.

Таблица 7d

Регистр границы вторичного входного ATU - SIALR

Бит	По умолчанию	Считывание/ запись	Описание
31:04	0000000H	Только считывание	Вторичная граница входного ATU - Это значение считы- вания только что записан- ной информации, которое определяет размер блока памяти, требуемого для устройства преобразования вторичного ATU
03:00	0000 <sub>2</sub>	Только считывание	Зарезервирован

RU 2157000 C2

RU 2157000 C2

Таблица 7е

Регистр значения преобразования вторичного входного  
 ATU - SIATVR

Бит	По умолчанию	Считывание/ запись	Описание
31:02	0000.0800H	Только считывание	Значение преобразования вторичного входного ATU - Это значение используется для преобразования адреса вторичной PCI-шины в лока- льный адрес. Значение пре- образования вторичного входящего адреса должно быть словом, выровненным по локальной шине. Адрес по умолчанию должен разрешать обращение устройства преоб- разования к внутренним P2P регистрам.
01:00	00 <sub>2</sub>	Только считывание	Зарезервирован

RU 2157000 C2

RU 2157000 C2

Таблица 7f

Первичный выходной регистр значения окна памяти - POMWVR

Бит	По умолчанию	Считывание/ запись	Описание
31:02	00000000H	Считывание/ запись	Значение первичного выходного MW - Это значение используется для преобразования локального адреса в PCI-адреса.
01:00	00 <sub>2</sub>	Только считывание	Порядок разбивки - это битовое поле указывает адресную последовательность во время разбивки памяти. Все адресаты должны проверять состояние нулевых и единичных адресных битов для определения последовательности адресации памяти, которую инициатор предназначает для использования во время пакетной передачи. (00 <sub>2</sub> ) Линейная, или последовательная, последовательность адресации во время разбивки.

RU 2157000 C2

RU 2157000 C2

Первичный выходной регистр значения окна ввода/вывода -

POIOWVR

Бит	По умолчанию	Считывание/ запись	Описание
31:02	0000.0000H	Считывание/ запись	Первичное выходное значение окна ввода/вывода - Это значение используется для преобразования локальных адресов в PCI-адреса.
01:00	00 <sub>2</sub>	Только считывание	Порядок разбивки - это битовое поле указывает адресную последовательность во время разбивки памяти. Все адресаты должны проверять состояние нулевых и единичных адресных битов для определения последовательности адресации памяти, которую инициатор предназначает для использования во время пакетной передачи, (00 <sub>2</sub> ) Линейная, или последовательная последовательность адресации во время разбивки

RU 2157000 C2

RU 2157000 C2

## Первичный выходной регистр значения окна DAC - PODWVR

Бит	По умолчанию	Считывание/ запись	Описание
31:02	00000000H	Считывание/ запись	Первичное выходное значение окна DAC - Это значение первичное ATU используется для преобразования локального адреса в PCI-адреса.
01:00	00 <sub>2</sub>	Только считывание	Порядок разбивки - это битовое поле указывает адресную последовательность во время разбивки памяти. Все адресаты должны проверять состояние нулевых и единичных адресных битов для определения последовательности адресации памяти, которую инициатор предназначает для использования во время пакетной передачи. (00 <sub>2</sub> ) Линейная, или последовательная, последовательность адресации во время разбивки.

RU 2157000 C2

RU 2157000 C2

Таблица 7i

Первичный выходной регистр верхних битов 64-битного

DAC - POU DR

Бит	По умолчанию	Считывание/ запись	Описание
31:00	0000.0000H	Считывание/ запись	Эти биты определяют верхние 32 бита адреса, управляемого во время двойного адресного цикла (DAC) .

RU 2157000 C2

RU 2157000 C2



Таблица 7j

Вторичный выходной регистр значения окна памяти - SOMWVR

Бит	По умолчанию	Считывание/ запись	Описание
31:02	0000.0000H	Считывание/ запись	Вторичное выходное значение окна памяти - Это значение используется для преобразования локальных адресов в PCI-адреса.
01:00	00 <sub>2</sub>	Только считывание	Порядок разбивки - это битовое поле указывает адресную последовательность во время разбивки памяти. Все адресаты должны проверять состояние нулевых и единичных адресных битов для определения последовательности адресации памяти, которую инициатор предназначен для использования во время пакетной передачи.  (00 <sub>2</sub> ) Линейная, или последовательная, последовательность адресации во время разбивки.

RU 2157000 C2

RU 2157000 C2

Вторичный выходной регистр значения окна ввода/вывода -  
SOIOWVR

Бит	По умолчанию	Считывание/ запись	Описание
31:02	0000.0000H	Считывание/ запись	Вторичное выходное значение окна ввода/вывода - Это зна- чение используется для преобразования локальных ад- ресов в PCI-адреса.
01:00	00 <sub>2</sub>	Только считывание	Порядок разбивки - это би- товое поле указывает адресную последовательность во время разбивки памяти. Все адресаты должны прове- рять состояние нулевых и единичных адресных битов для определения последовательно- сти адресации памяти, которую инициатор предназна- чает для использования во время пакетной передачи. (00 <sub>2</sub> ) Линейная, или последо- вательная, последователь- ность адресации во время разбивки.

RU 2157000 C2

RU 2157000 C2

Таблица 78

Регистр границы расширенной памяти ROM - ERLR

Бит	По умолчанию	Считывание/ запись	Описание
31:11	000000H	Только считывание	Граница расширенной памяти ROM - размер блока памяти, требуемого для устройства преобразования расширенной памяти ROM. Значение по умолчанию, равное 0, означает, что адресное пространство не является расширенной памятью ROM.
01:00	00 <sub>2</sub>	Только считывание	Зарезервировано

RU 2157000 C2

RU 2157000 C2

Регистр значения преобразования расширенной памяти ROM -  
ERTVR

Бит	По умолчанию	Считывание/ запись	Описание
31:02	00000000H	Только считывание	Значение преобразования рас- ширенной памяти ROM локального процессора - Это значение ис- пользуется для преобразования PCI-адресов в локальные адреса для обращения к расширенной памяти ROM.  Значение преобразования адреса расширенной памяти ROM должно быть словом, выровненным по локальной шине.
01:00	00 <sub>2</sub>	Только считывание	Зарезервировано

RU 2157000 C2

RU 2157000 C2

## Регистр конфигурации ATU - ATUCR

Бит	По умолчанию	Считывание/ запись	Описание
31:12	00000H	Только считывание	Зарезервировано
11	0 <sub>2</sub>	Считывание/ запись	Режим загрузки вторичной PCI- Если установлен, то вторичное ATU будет востребовать об- ращения ко всей локальной ши- не с адресами, находящимися в диапазоне FE000000H-FFFFFFFFH. Это позволяет локальному про- цессору самозагружаться со вторичной PCI-шины. Алгоритм преобразования будет исполь- зовать в этом режиме вторичный выходной регистр значения окна ввода/вывода.

RU 2157000 C2

RU 2157000 C2

10	0 <sub>2</sub>	Только считывание	Зарезервировано
09	0 <sub>2</sub>	Только считывание	Зарезервировано
08	0 <sub>2</sub>	Считывание/ запись	Разрешение прямой адресации - Установка этого бита разрешает прямую адресацию через устрой- ства ATU. Циклы локальной шины с адресами между 0000.1000H и 07FFF.FFFFH должны быть авто- матически переданы на PCI-шину без какого-либо преобразования адреса. ATU, которое требует прямого преобразования адреса- ции, зависит от бита выбора прямой адресации вторичной шины.
07	0 <sub>2</sub>	Считывание/ запись	Выбор прямой адресации вторич- ной шины - Установка этого бита приводит к прямой адреса- ции выходящих транзакций, которые должны быть переданы через вторичное ATU во вторичную PCI-шину. Когда бит очищен, прямая адресация испо- льзует первичное ATU и первичную PCI-шину. Бит разре-

			шения прямой адресации должен быть установлен для разрешения прямой адресации.
06	0 <sub>2</sub>	Считывание/ запись	Разрядность расширенной ROM - когда очищен, этот бит означает, что используется 8-битовая расширенная ROM. Когда установлен, этот бит означает, что используется 32-битовая расширенная ROM. Этот бит используется совместно с разрешением декодирования адреса (бит 0) регистра ERBAR.
05	0 <sub>2</sub>	Считывание/ запись	Разрешение прерывания по сигналу об ошибке PCI вторичного ATU - Этот бит действует как маска для битов 4:0 регистра состояния прерывания вторичного ATU. Установка этого бита разрешает прерывание к локальному процессору, когда любой из этих битов установлен в SATUISR. Очистка этого бита блокирует прерывание.
04	0 <sub>2</sub>	Считывание/ запись	Разрешение прерывания по сигналу об ошибке PCI первичного ATU - Этот бит действует как

			маска для битов 4:0 регистра состояния прерывания первичного ATU. Установка этого бита разрешает прерывание к локальному процессору, когда любой из этих битов установлен в PATUISR. Очистка этого бита блокирует прерывание.
03	0 <sub>2</sub>	Считывание/ запись	Разрешение прерывания ATU от BIST (встроенного самоконтроля) - Установка этого бита разрешает прерывание к локальному процессору, когда начальный бит BIST установлен в регистре ATUBISTR. Этот бит также является скопированным как бит 7 возможности BIST в регистре ATUBISTR.
02	0 <sub>2</sub>	Считывание/ запись	Разрешение вторичного выходного ATU - Установка этого бита разблокирует вторичное выходное устройство преобразования адреса.  Очистка блокирует ATU.
01	0 <sub>2</sub>	Считывание/ запись	Разрешение первичного выходного ATU - Установка этого бита разблокирует первичное выход-



			ное устройство преобразования адреса. Очистка блокирует первичное выходное ATU.
00	0 <sub>2</sub>	Считывание/ запись	Разрешение прерывания входного "дверного колокольчика" - Если установлен, то этот бит разрешает устройству сообщений генерировать прерывание локального процессора для каждой операции записи во входной регистр "дверного колокольчика". Если очищен, то никакого прерывания не генерируется.

## Регистр состояния прерывания первичного ATU - PATUISR

Бит	По умолчанию	Считывание/ запись	Описание
31:11	000000H	Только считывание	Зарезервировано
10	0 <sub>2</sub>	Только считывание	Зарезервировано
9	0 <sub>2</sub>	Только считывание	Зарезервировано
8	0 <sub>2</sub>	Считывание/ очистка	Прерывание ATU от BIST (встроенный самоконтроль). Если этот бит установлен, то главный процессор производит начальный BIST, бит 6 регистра ATUBISTR, разрешено прерывание ATU от BIST, бит 12 регистра ATUCR является разрешенным. Локальный процессор может инициировать программный BIST и сохранить результат в битах

			[3:0] регистра ATUBISTR.
07	0 <sub>2</sub>	Только считывание	Зарезервировано
06	0 <sub>2</sub>	Считывание/очистка	Сбой памяти локального процессора - Этот бит является установленным, когда контроллер памяти определяет сбой памяти, а первичное ATU было ведущим устройством для транзакции.
05	0 <sub>2</sub>	Считывание/очистка	Сбой шины локального процессора - Этот бит является установленным, когда контроллер памяти определяет сбой шины, а первичное ATU было ведущим устройством для транзакции.
04	0 <sub>2</sub>	Считывание/очистка	Установлен уровень сигнала P_SERR# - этот бит устанавливается всякий раз когда уровень сигнала P_SERR# устанавливается на шине PCI.
03	0 <sub>2</sub>	Считывание/очистка	Преждевременное прерывание работы ведущего PCI- этот бит является установленным всякий раз, когда транзакция, инициированная интерфейсом ведущего ATU, заканчивается в преждевременно прерванном ведущем

			устройстве.
02	0 <sub>2</sub>	Считывание/ очистка	Преждевременное прерывание работы ведущего PCI (ведущего) - этот бит устанавливается всякий раз, когда транзакция, инициированная интерфейсом ведущего АТУ, заканчивается в преждевременно прерванном ведущем устройстве .
01	0 <sub>2</sub>	Считывание/ очистка	Преждевременное прерывание работы ведущего PCI (адресата) - этот бит устанавливается всякий раз, когда интерфейс АТУ, выступающий как адресат, прерывает транзакцию на PCI-шине со сбросом адресата.
00	0 <sub>2</sub>	Считывание/ запись	Ошибка четности ведущего PCI - интерфейс АТУ устанавливает этот бит, когда удовлетворяются три условия: 1)исполнительное устройство шины устанавливает уровень сигнала P_PERR# самостоятельно или обнаруживает установленный уровень сигнала P_PERR#, 2)исполнительное устройство, устанавливающее бит, действует как "хозяин"

			шины для операции, в которой имеет место ошибка, 3) бит отклика на ошибку четности (регистр команд) установлен. Этот бит очищается главным процессором, выполняющим операцию считывания/очистки над битом 8 регистра PATUSR.
--	--	--	--

Таблица 7р

Регистр состояния прерывания вторичного ATU - SATUISR

Бит	По умолчанию	Считывание/ запись	Описание
31:07	0000000H	Только Считывание	Зарезервировано
06	0 <sub>2</sub>	Считывание/	Сбой памяти локального про-

RU 2157000 C2

RU 2157000 C2

		очистка	цессора - Этот бит является установленным, когда контроллер памяти определяет сбой памяти, а вторичное ATU было ведущим устройством для транзакции.
05	0 <sub>2</sub>	Считывание/ очистка	Сбой шины локального процессора - Этот бит является установленным, когда контроллер памяти определяет сбой шины, а вторичное ATU было ведущим устройством для транзакции.
04	0 <sub>2</sub>	Считывание/ очистка	Установлен уровень сигнала P_SERR# - этот бит устанавливается всякий раз, когда уровень сигнала P_SERR# устанавливается на шине PCI.
03	0 <sub>2</sub>	Считывание/ очистка	Преждевременное прерывание работы ведущего PCI - этот бит является установленным всякий раз, когда транзакция, инициированная интерфейсом ведущего ATU, заканчивается в преждевременно прерванном ведущем устройстве.
02	0 <sub>2</sub>	Считывание/ очистка	Преждевременное прерывание работы ведущего PCI (ведуще-

			го) - этот бит устанавливается всякий раз, когда транзакция, инициированная интерфейсом ведущего АТУ, заканчивается в преждевременно прерванном ведущем устройстве.
01	0 <sub>2</sub>	Считывание/ запись	Преждевременное прерывание работы ведущего РСІ (адресата) - этот бит устанавливается всякий раз, когда интерфейс АТУ, выступающий как адресат, преждевременно прерывает транзакцию на РСІ-шине со сбросом адресата.
00	0 <sub>2</sub>	Считывание/ запись	Ошибка четности ведущего РСІ-интерфейс АТУ устанавливает этот бит, когда удовлетворяются три условия: 1) исполнительное устройство шины устанавливает уровень сигнала P_PERR# самостоятельно или обнаруживает установленный уровень сигнала P_PERR#, 2) исполнительное устройство, устанавливающее бит, действует как "хозяин" шины для операции, в которой имеет место

			ошибка, 3) бит отклика на ошибку четности (регистр команд) установлен. Этот бит очищается главным процессором, выполняющим операцию считывания/очистки над битом 8 регистра PATUSR.
--	--	--	---

Таблица 7q

Регистр команд вторичного ATU - SATUCMD

Бит	По умолчанию	Считывание/запись	Описание
31:10	000000H	Только считывание	Зарезервировано
09	0 <sub>2</sub>	Считывание/запись	Разрешение быстрой передачи "туда-обратно" - Если этот бит очищен, то интерфейсу вторичного ATU не разрешено генерировать циклы быстрой передачи "туда-обратно" на своей шине.

RU 2157000 C2

RU 2157000 C2



08	0 <sub>2</sub>	Считывание/ запись	Разрешение S_SERR# - Если этот бит очищен, интерфейсу вторичного ATU не разрешено устанавливать уровень сигнала S_SERR# на интерфейсе PCI.
07	0 <sub>2</sub>	Только считывание	Управление циклом ожидания - управляет пошаговым выполнением команд адрес/данные. Не реализовано и битовое поле зарезервировано.
06	0 <sub>2</sub>	Считывание/ запись	Разрешение проверки четности - этот бит установлен, то интерфейс вторичного ATU должен предпринять обычные действия при определении ошибки четности. Если бит очищен, проверка четности блокируется.
05	0 <sub>2</sub>	Только считывание	Разрешение просмотра палитры VGA - интерфейс вторичного ATU не поддерживает операции ввода/вывода и поэтому не выполняет просмотр палитры VGA.
04	0 <sub>2</sub>	Только считывание	Разрешение записи и недостоверные данные - не применимо. Не реализовано и битовое поле зарезервировано.

03	0 <sub>2</sub>	Только считывание	Разрешение специального цикла - интерфейс ATU не откликается на специальные циклические команды в любом случае. Не применимо. Не реализовано и битовое поле зарезервировано.
02	0 <sub>2</sub>	Считывание/запись	Разрешение "хозяина" шины-интерфейс вторичного ATU имеет способность выступать в качестве ведущего устройства на шине PCI. Значение 0 блокирует устройство от генерирования обращений к PCI. Значение 1 разрешает устройству выступать как "хозяин" шины. Этот бит разрешения также управляет ведущим интерфейсом каналов DMA 0 и 1. Бит должен быть установлен прежде инициализации передачи с использованием DMA на шине PCI.
01	0 <sub>2</sub>	Считывание/запись	Разрешение памяти - управляет откликом интерфейса вторичного ATU к адресам памяти PCI. Если этот бит очищен, интерфейс ATU не откликается на любые обращения к памяти на

			шине PCI.
00	0 <sub>2</sub>	Только считывание	Разрешение пространства ввода/вывода - управляет откликом интерфейса вторичного АТУ на транзакции ввода/вывода на первичной стороне. Не реализовано и битовое поле зарезервировано.

Таблица 7 г

Вторичный выходной регистр значения окна DAC - SODWVR

Бит	По умолчанию	Считывание/ запись	Описание
31:02	00000000H	Считывание/ запись	Вторичное выходное значение окна DAC - Это значение вторичное АТУ использует для преобразования локального адреса в PCI-адреса.
01:00	00 <sub>2</sub>	Только считывание	Порядок разбивки - это битовое поле указывает адресную последовательность во время разбивки памяти. Все адресаты должны проверять состояние нулевых и единичных адресных битов для определения последовательности адресации памяти, которую инициатор предназначает для использования во время пакетной передачи. (00 <sub>2</sub> ) Линейная, или последовательная, последовательность адресации во время разбивки.

RU 2157000 C2

RU 2157000 C2

Таблица 7s

Вторичный выходной регистр верхних битов 64-битового DAC -  
SOUDR

Бит	По умолчанию	Считывание/ запись	Описание
31:00	0000.0000H	Считывание/ запись	Вторичный выходной верхний 64-битовый DAC адрес - эти биты определяют верхние 32 бита адреса, управляемого во время двойного адресного цикла (DAC).

Таблица 7s'

Первичный выходной адресный регистр цикла конфигурации -  
POCCAR

Бит	По умолчанию	Считывание/ запись	Описание
31:00	0000.0000H	Только считывание	Первичный адрес цикла конфигурации - эти биты определяют 32-битный адрес, используемый во время цикла считывания или записи выходной конфигурации.

RU 2157000 C2

RU 2157000 C2

Таблица 7u

Вторичный выходной адресный регистр цикла конфигурации -  
SOCCAR

Бит	По умолчанию	Считывание/ запись	Описание
31:00	0000.0000H	Только считывание	Вторичный адрес цикла конфигурации - эти биты определяют 32-битный адрес, используемый во время цикла считывания или записи выходной конфигурации.

RU 2157000 C2

RU 2157000 C2

Таблица 8

Механизм	Количество	Устанавливает уровни сигналов прерывания PCI?	Формирует прерывание локального процессора?
Регистры сообщений	2 входных 2 выходных	Да	Необязательно
Регистры "дверного колокольчика"	1 входной 1 выходной	Да	Необязательно
Циклические очереди	4 циклических очереди	При некоторых условиях	При некоторых условиях
Индексные регистры	1004 32-битные области памяти	Нет	Необязательно
Регистры APIC	1 регистр выбор 1 окно	Нет	Да

RU 2157000 C2

RU 2157000 C2

## Обзор программируемого контроллера прерывания P2P

Режим контроллера прерываний локального процессора									
Назначе- нный режим	Расшир- енный режим	Смешан- ный режим	ICON. im	Прерыв- ание, фиксир- уемое по уро- вню	Прерыв- ание, фиксир- уемое по фро- нту	ICON. sdm	Ре- жим выбо- рки	ICON . sm	Пре- рыва- ние лока- льно- го проце- ссора
Разре- шено	Блоки- ровано	Блоки- ровано	00 <sub>2</sub>	Разре- шено	Блоки- ровано	0 <sub>2</sub>	Быстр- ый	1 <sub>2</sub>	XINT0 #/INTA#
Разре- шено	Блоки- ровано	Блоки- ровано	00 <sub>2</sub>	Разре- шено	Блоки- ровано	0 <sub>2</sub>	Быстр- ый	1 <sub>2</sub>	XINT1 #/INTB#
Разре- шено	Блоки- ровано	Блоки- ровано	00 <sub>2</sub>	Разре- шено	Блоки- ровано	0 <sub>2</sub>	Быстр- ый	1 <sub>2</sub>	XINT2 #/INIC

RU 2157000 C2

RU 2157000 C2



RU 2157000 C2

									INTC#
Разре - шено	Блоки - ровано	Блоки - ровано	00 <sub>2</sub>	Разре - шено	Блоки - ровано	0 <sub>2</sub>	Быстр - ый	1 <sub>2</sub>	XINT3 #/ INTD#
Разре - шено	Блоки - ровано	Блоки - ровано	00 <sub>2</sub>	Разре - шено	Блоки - ровано	0 <sub>2</sub>	Быстр - ый	1 <sub>2</sub>	XINT4 #
Разре - шено	Блоки - ровано	Блоки - ровано	00 <sub>2</sub>	Разре - шено	Блоки - ровано	0 <sub>2</sub>	Быстр - ый	1 <sub>2</sub>	XINT5 #
Разре - шено	Блоки - ровано	Блоки - ровано	00 <sub>2</sub>	Разре - шено	Блоки - ровано	0 <sub>2</sub>	Быстр - ый	1 <sub>2</sub>	XINT6 #
Разре - шено	Блоки - ровано	Блоки - ровано	00 <sub>2</sub>	Разре - шено	Блоки - ровано	0 <sub>2</sub>	Быстр - ый	1 <sub>2</sub>	XINT7 #
Не досту - пен	Не досту - пен	Не досту - пен	Не досту - пен	Не досту - пен	По умол - чанию	Не досту - пен	Быстр - ый	1 <sub>2</sub>	NMI#

RU 2157000 C2

## Маршрутизация прерывания PCI

Бит выбора XINT	Описание
0	<p>Входная линия INTA#/XINT0#, направляемая на входную линию XINT0# локального процессора</p> <p>Входная линия INTB#/XINT1#, направляемая на входную линию XINT1# локального процессора</p> <p>Входная линия INTC#/XINT2#, направляемая на входную линию XINT2# локального процессора</p> <p>Входная линия INTD#/XINT3#, направляемая на входную линию XINT3# локального процессора</p>
1	<p>Входная линия INTA#/XINT0#, направляемая на выходную линию P_INTA#</p> <p>Входная линия INTB#/XINT1#, направляемая на выходную линию P_INTB#</p> <p>Входная линия INTC#/XINT2#, направляемая на выходную линию P_INTC#</p> <p>Входная линия INTD#/XINT3#, направляемая на выходную линию P_INTD#</p>

RU 2157000 C2

RU 2157000 C2

Источники прерывания XINT6#

Устройство	Условие прерывания	Регистр	Положение бита
DMA канал 0	Конец цепочки	Регистр состояния канала 0	08
	Конец передачи	Регистр состояния канала 0	09
DMA канал 1	Конец цепочки	Регистр состояния канала 1	08
	Конец передачи	Регистр состояния канала 1	09
DMA канал 2	Конец цепочки	Регистр состояния канала 2	08
	Конец передачи	Регистр состояния канала 2	09

RU 2157000 C2

RU 2157000 C2

## Источники прерывания XINT7#

Устройство	Условие прерывания	Регистр	Положение бита
Интерфейсное устройство шины APIC	Сообщение EOI принято	Регистр состояния/управления APIC	14
	Сообщение APIC послано	Регистр состояния/управления APIC	06
Интерфейсное устройство шины I <sup>2</sup> C	Приемный буфер заполнен	Регистр состояния I <sup>2</sup> C	07
	Передающий буфер пуст	Регистр состояния I <sup>2</sup> C	06
	Адрес подчиненного устройства определен	Регистр состояния I <sup>2</sup> C	05
	Определен сигнал STOP	Регистр состояния I <sup>2</sup> C	04
	Определена ошибка шины	Регистр состояния I <sup>2</sup> C	03
	Определена потеря арбитража	Регистр состояния I <sup>2</sup> C	02
Устройство передачи сообщений	Прерывание "дверного колокольчика"	Регистр состояния прерывания первичного ATU	09
Первичное ATU	Начало BIST ATU	Регистр состояния прерывания первичного ATU	08

Источники прерывания NMI

Устройство	Условие прерывания	Регистр	Положение бита
Мостовой интерфейс	Ошибка четности	Регистр состояния	00

RU 2157000 C2

RU 2157000 C2

первичного PCI	ведущей PCI	прерывания первичного моста	
	Преждевременное прерывание работы адресата PCI (адресат)	Регистр состояния прерывания первичного моста	01
	Преждевременное прерывание работы адресата PCI (ведущее устройство)	Регистр состояния прерывания первичного моста	02
	Преждевременное прерывание работы ведущей PCI	Регистр состояния прерывания первичного моста	03
	Установление уровня сигнала P_SERR#	Регистр состояния прерывания первичного моста	04
Мостовой	Ошибка	Регистр	00

интерфейс вторичной PCI	четности ведущей PCI	состояния прерывания вторичного моста	01
	Преждевремен ное прерывание работы адресата PCI (адресат)	Регистр состояния прерывания вторичного моста	
	Преждевремен ное прерывание работы адресата PCI (ведущее устройство)	Регистр состояния прерывания вторичного моста	
	Преждевремен ное прерывание работы ведущей PCI	Регистр состояния прерывания вторичного моста	
	Установление уровня сигнала S_SERR#	Регистр состояния прерывания вторичного моста	04

Первичное АТУ	Ошибка четности ведущей PCI	Регистр состояния прерывания первичного АТУ	00
	Преждевремен ное прерывание работы адресата PCI (адресат)	Регистр состояния прерывания первичного АТУ	01
	Преждевремен ное прерывание работы адресата PCI (ведущее устройство)	Регистр состояния прерывания первичного АТУ	02
	Преждевремен ное прерывание работы ведущей PCI	Регистр состояния прерывания первичного АТУ	03
	Установление уровня сигнала P_SERR#	Регистр состояния прерывания первичного	04



		АТУ	
	Сбой шины локального процессора	Регистр состояния прерывания первичного АТУ	05
	Сбой памяти локального процессора	Регистр состояния прерывания первичного АТУ	06
Вторичное АТУ	Ошибка четности ведущей PCI	Регистр состояния прерывания вторичного АТУ	00
	Преждевремен ное прерывание работы адресата PCI (адресат)	Регистр состояния прерывания вторичного АТУ	01
	Преждевремен ное прерывание работы адресата PCI	Регистр состояния прерывания вторичного АТУ	02

	(ведущее устройство)		
	Преждевременное прерывание работы ведущей PCI	Регистр состояния прерывания вторичного ATU	03
	Установление уровня сигнала S_SERR#	Регистр состояния прерывания вторичного ATU	04
	Сбой шины локального процессора	Регистр состояния прерывания вторичного ATU	05
	Сбой памяти локального процессора	Регистр состояния прерывания вторичного ATU	06
Локальный процессор	Сбой шины локального процессора	Регистр состояния локального процессора	05
	Сбой памяти	Регистр	06

	локального процессора	состояния локального процессора	
Канал DMA 0	Ошибка четности ведущей PCI	Регистр состояния канала 0	00
	Преждевремен ное прерывание работы адресата PCI (ведущее устройство)	Регистр состояния канала 0	02
	Преждевремен ное прерывание работы ведущей PCI	Регистр состояния канала 0	03
	Сбой шины локального процессора	Регистр состояния канала 0	05
	Сбой памяти локального процессора	Регистр состояния канала 0	06
Канал DMA1	Ошибка четности ведущей PCI	Регистр состояния канала 1	00

	Преждевременное прерывание работы адресата PCI (ведущее устройство)	Регистр состояния канала 1	02
	Преждевременное прерывание работы ведущей PCI	Регистр состояния канала 1	03
	Сбой шины локального процессора	Регистр состояния канала 1	05
	Сбой памяти локального процессора	Регистр состояния канала 1	06
Канал DMA2	Ошибка четности ведущей PCI	Регистр состояния канала 2	00
	Преждевременное прерывание работы адресата PCI (ведущее	Регистр состояния канала 2	02

устройство) Преждевременное прерывание работы ведущей PCI	Регистр состояния канала 2	03
	Регистр состояния канала 2	05
	Регистр состояния канала 2	06

Таблица 14

Описание входных линий прерывания

Сигнал	Описание
INTA#/XINT0#	Это линия прерывания может быть направлена на вывод P_INTA# или вход XINT0 прерывания локального процессора.
INTB#/XINT1#	Это линия прерывания может быть направлена на вывод P_INTB# или вход

	XINT1 прерывания локального процессора.
INTC#/XINT2#	Это линия прерывания может быть направлена на вывод P_INTC# или вход XINT2 прерывания локального процессора.
INTD#/XINT3#	Это линия прерывания может быть направлена на вывод P_INTD# или вход XINT3 прерывания локального процессора.
XINT4#	Этот вход прерывания всегда соединен со входом XINT4 прерывания локального процессора
XINT5#	Этот вход прерывания всегда соединен со входом XINT5 прерывания локального процессора
XINT6#	Этот вход прерывания совместно используется тремя внутренними прерываниями. Ими являются прерывания от каждого из внутренних каналов DMA. Все прерывания направлены на вход локального процессора XINT6#. Программное обеспечение должно считывать регистр состояния прерывания XINT6 для определения точного источника прерывания.
XINT7#	Этот вход прерывания совместно используется четырьмя внутренними прерываниями. Ими являются прерывания от интерфейсного устройства шины APIC, интерфейсного устройства шины I <sup>2</sup> C,

	<p>первичного ATU и устройства передачи сообщений. Все прерывания направлены на вход локального процессора XINT7#.</p> <p>Программное обеспечение должно считывать регистр состояния прерывания XINT7 для определения точного источника прерывания.</p>
NMI#	<p>Этот вход прерывания совместно используется восемью внутренними прерываниями. Они включают прерывание объединенной ошибки от локального процессора, мостового интерфейса первичного PCI, мостового интерфейса вторичного PCI, первичного ATU, вторичного ATU, и трех DMA каналов. Все прерывания направлены на вход локального процессора NMI#. Программное обеспечение должно считывать регистр состояния прерывания NMI для определения точного источника прерывания.</p>

Регистр состояния прерывания XINT6.

Бит	По умолчанию	Считывание/ запись	Описание
31 : 04	0000000H	Только считывание	Зарезервирован
03	0 <sub>2</sub>	Только считывание	Задержка прерывания внешнего XINT6#-когда установлен, то прерывание является задержанным на внешнем входе XINT6#. Когда очищен, никакого прерывания не существует.
02	0 <sub>2</sub>	Только считывание	Задержка прерывания DMA канала 2-когда установлен, то DMA канал 2 указывает активное условие канала "конец цепочки". Когда очищен, никакого условия прерывания не существует.
01	0 <sub>2</sub>	Только считывание	Задержка прерывания DMA канала 1-когда установлен, то DMA канал 1 указывает активное условие канала "конец цепочки". Когда

RU 2157000 C2

RU 2157000 C2



			очищен, никакого условия прерывания не существует.
00	0 <sub>2</sub>	Только считывание	Задержка прерывания DMA канала 0-когда установлен, то DMA канал 0 указывает активное условие канала "конец цепочки". Когда очищен, никакого условия прерывания не существует.

Регистр состояния прерывания XINT7.

Бит	По умолчанию	Считывание/ запись	Описание
31:05	0000000H	Только считывание	Зарезервирован
04	0 <sub>2</sub>	Только считывание	Задержка прерывания внешнего XINT7#-когда установлен, то прерывание является задержанным на внешнем входе XINT7#. Когда очищен, никакого прерывания не существует.
03	0 <sub>2</sub>	Только считывание	Задержка прерывания первичного ATU/ начального BIST- -когда установлен, то главный процессор устанавливает запрос начального BIST в регистре

RU 2157000 C2

RU 2157000 C2

			ATUBISTR. Когда очищен, никакое прерывание начального BIST не задерживается.
02	0 <sub>2</sub>	Только считывание	Задержка прерывания входящего "дверного колокольчика"-когда установлен, то прерывание от входного устройства передачи сообщений задерживается. Когда очищен, никакое прерывание не задерживается.
01	0 <sub>2</sub>	Только считывание	Задержка прерывания от I <sup>2</sup> C - когда установлен, то прерывание от интерфейсного устройства шины I <sup>2</sup> C задерживается. Когда очищен, никакое прерывание не задерживается.
00	0 <sub>2</sub>	Только считывание	Задержка прерывания от APIC -когда установлен, то прерывание от интерфейсного устройства шины APIC задерживается. Когда очищен, никакое прерывание не задерживается.

## Регистр состояния прерывания NMI.

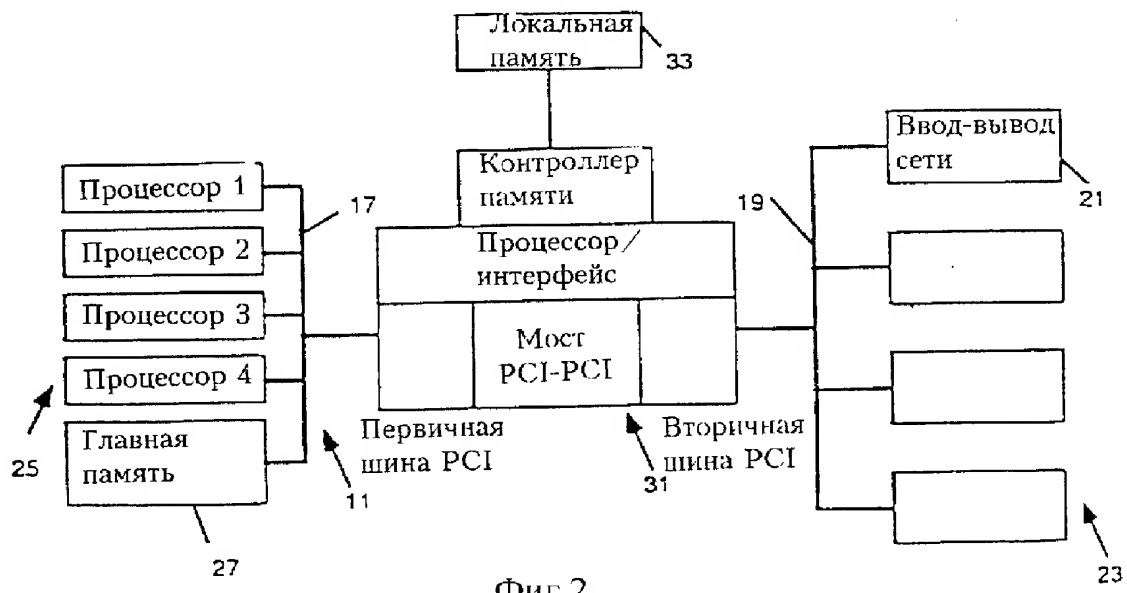
Бит	По умолчанию	Считывание/ запись	Описание
31:09	000000H	Только считывание	Зарезервирован
08	0 <sub>2</sub>	Только считывание	Прерывание внешнего NMI#- когда установлен, прерывание задерживается на внешнем входе NMI#. Когда очищен, никакого прерывания не существует.
07	0 <sub>2</sub>	Только считывание	Ошибка канала DMA 2- когда установлен, то существует условие ошибки PCI или локальной шины в канале DMA. Когда очищен, то никакой ошибки не существует.
06	0 <sub>2</sub>	Только считывание	Ошибка канала DMA 1- когда установлен, то существует условие ошибки PCI или локальной шины в канале DMA. Когда очищен, никакой ошибки не существует.
05	0 <sub>2</sub>	Только считывание	Ошибка канала DMA 0- когда установлен, то существует условие ошибки PCI или

RU 2157000 C2

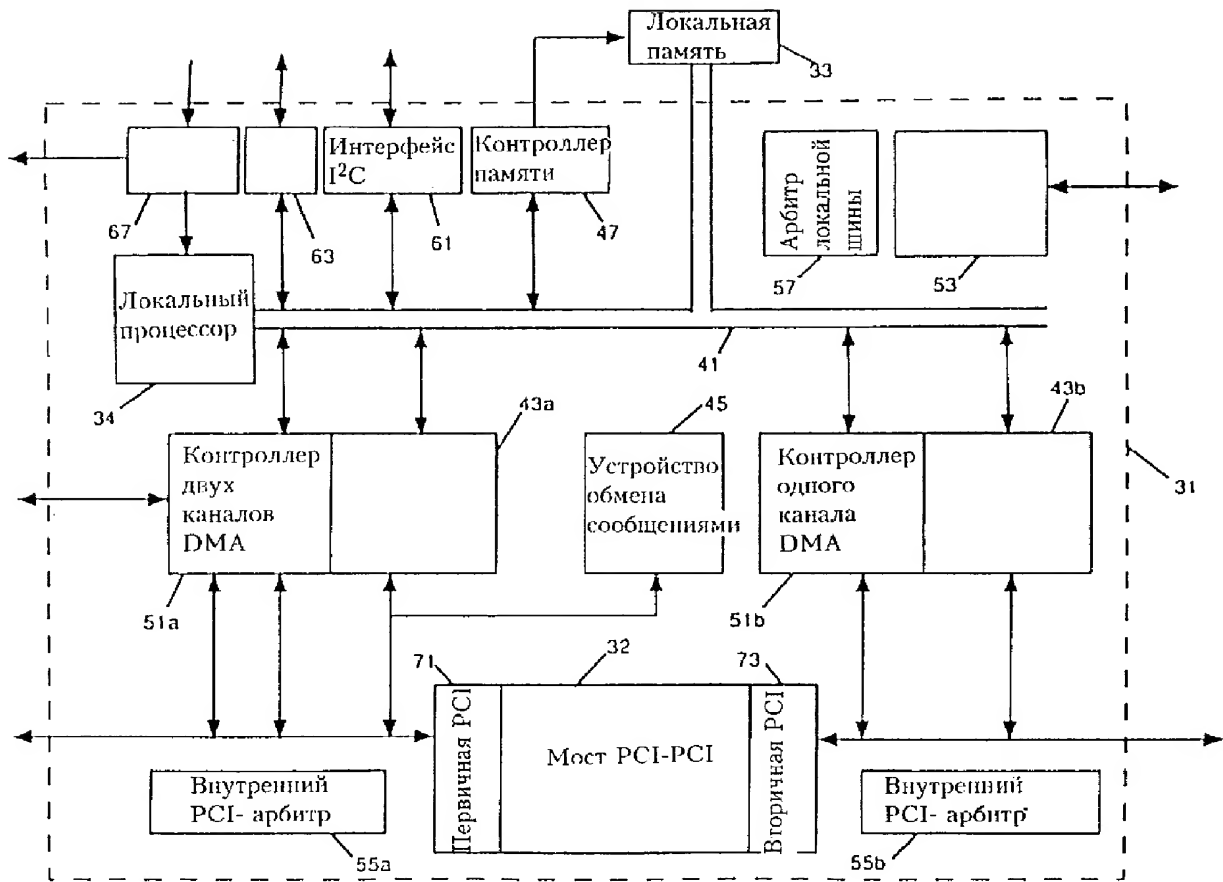
RU 2157000 C2

			локальной шины в канале DMA. Когда очищен, никакой ошибки не существует.
04	0 <sub>2</sub>	Только считывание	Ошибка вторичного моста- когда установлен, то существует условие ошибки PCI во вторичном интерфейсе моста. Когда очищен, никакой ошибки не существует.
03	0 <sub>2</sub>	Только считывание	Ошибка интерфейса первичного моста-когда установлен, то существует условие ошибки PCI в первичном интерфейсе моста. Когда очищен, то никакой ошибки не существует.
02	0 <sub>2</sub>	Только считывание	Ошибка вторичного ATU -когда установлен, то существует условие ошибки во вторичном ATU. Когда очищен, то ника- кой ошибки не существует.
01	0 <sub>2</sub>	Только считывание	Ошибка первичного ATU -когда установлен, то существует условие ошибки в первичном ATU. Когда очищен, то ника- кой ошибки не существует.
00	0 <sub>2</sub>	Только	Ошибка локального

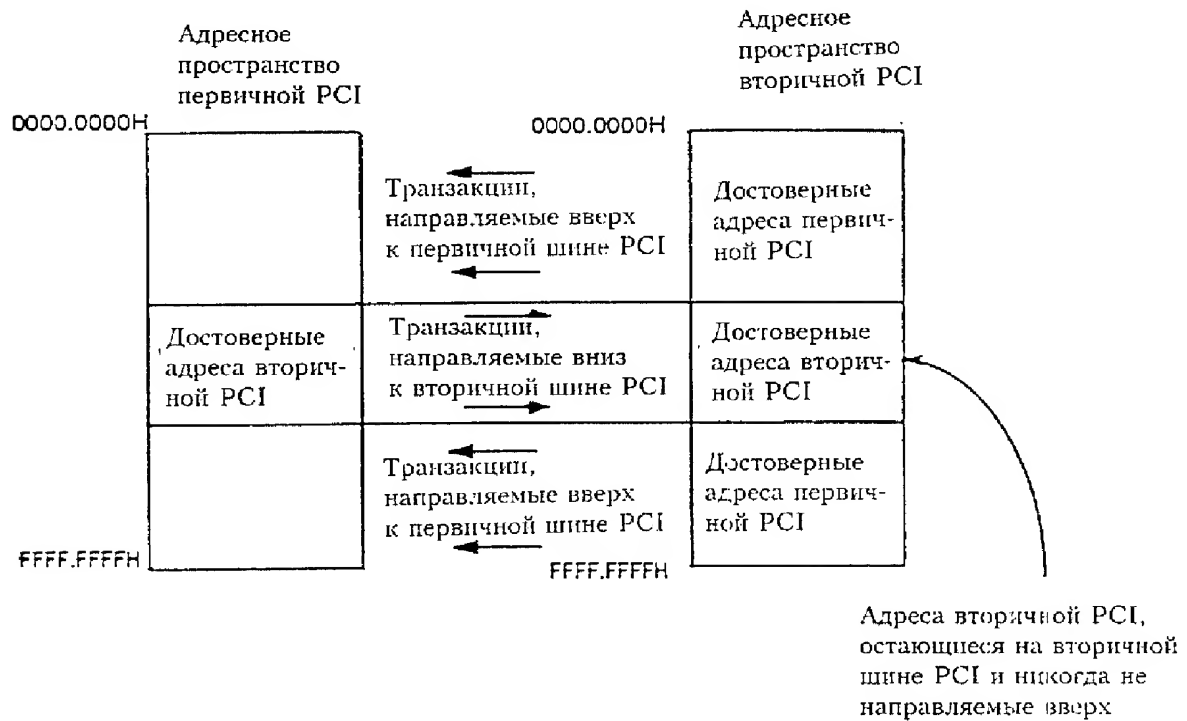
		считывание	процессора-когда установлен, то ошибочное условие, вызванное локальным процессором, существует во внутреннем контроллере памяти. Когда очищен, то никакой ошибки не существует.
--	--	------------	--



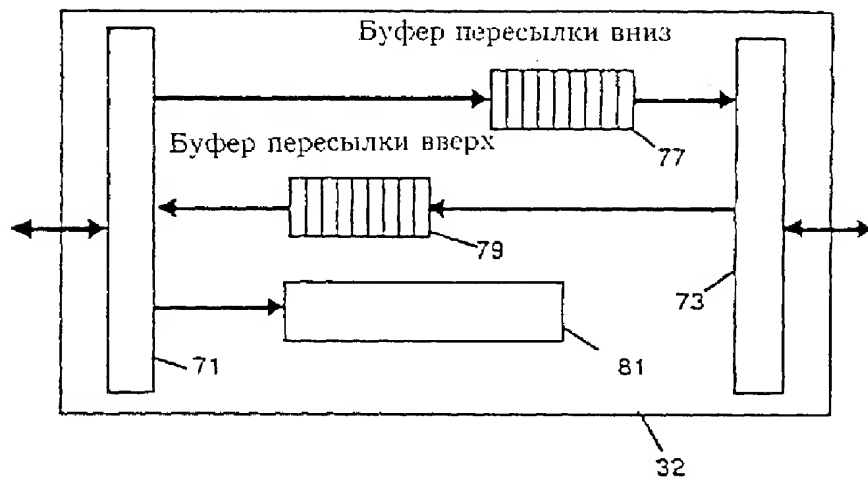
Фиг.2



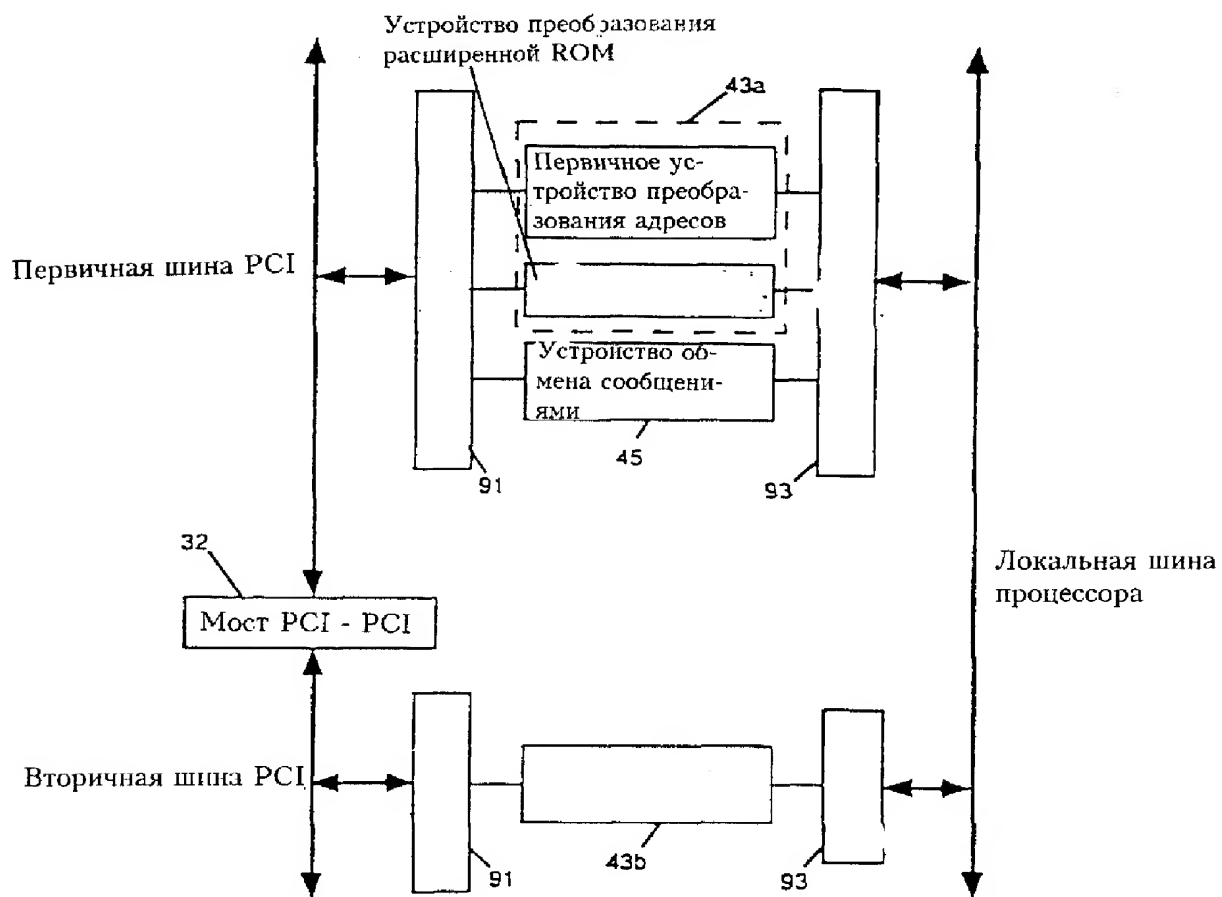
Фиг.3



Фиг.4

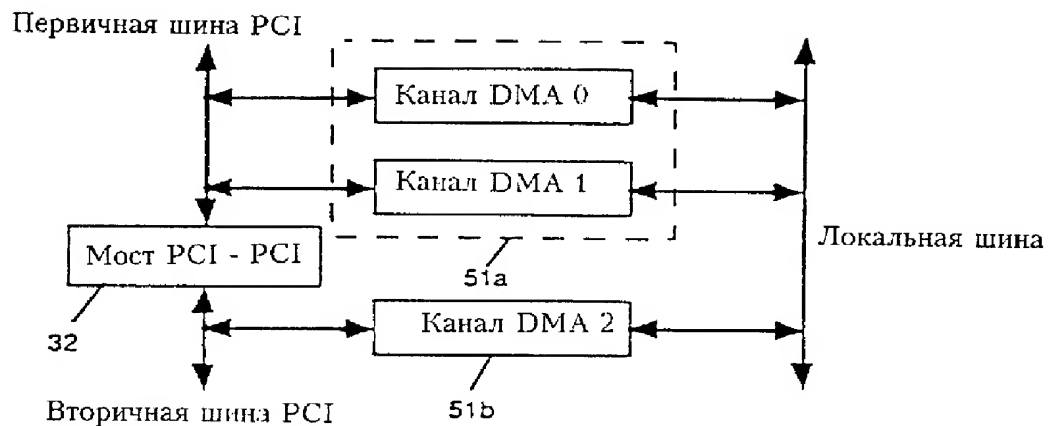


Фиг.5

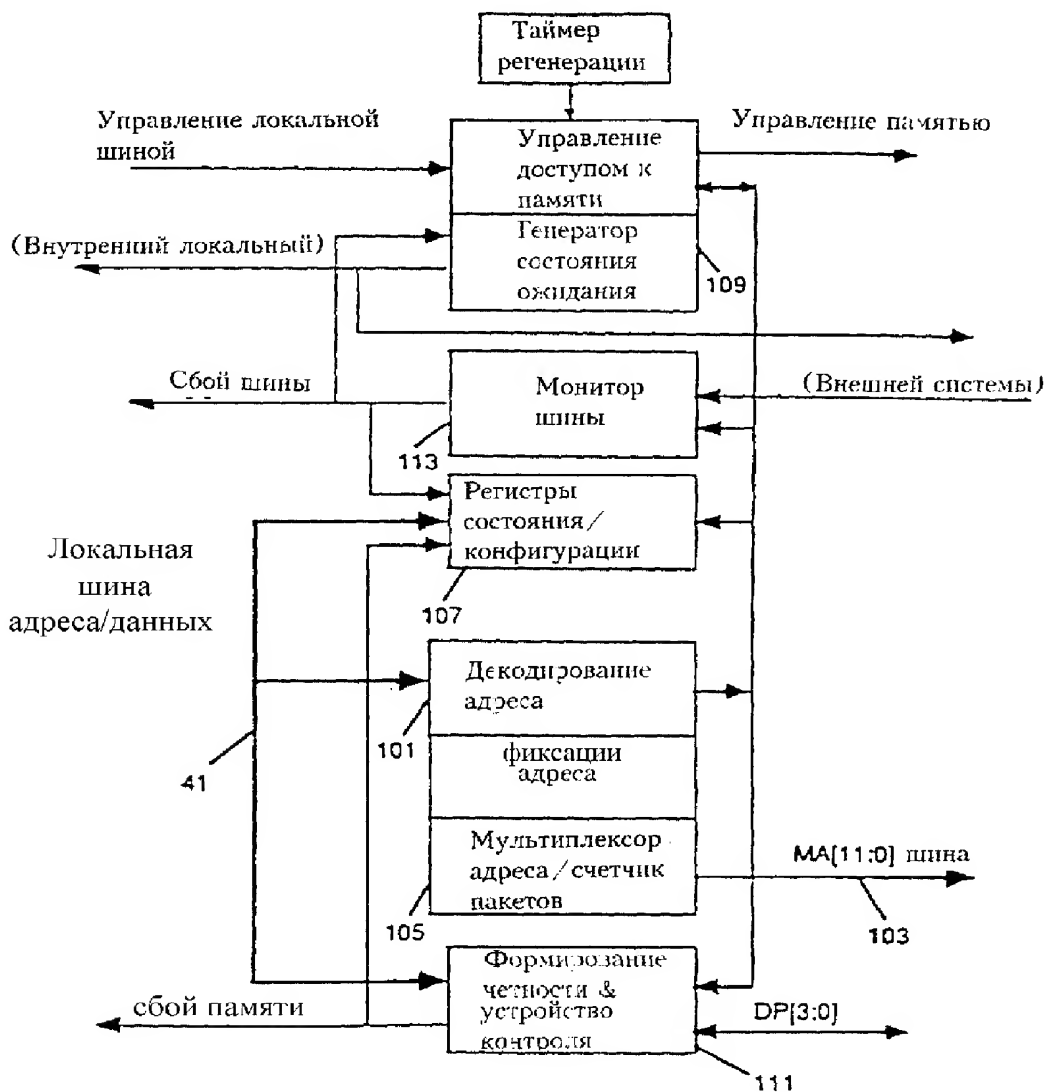


Фиг.6

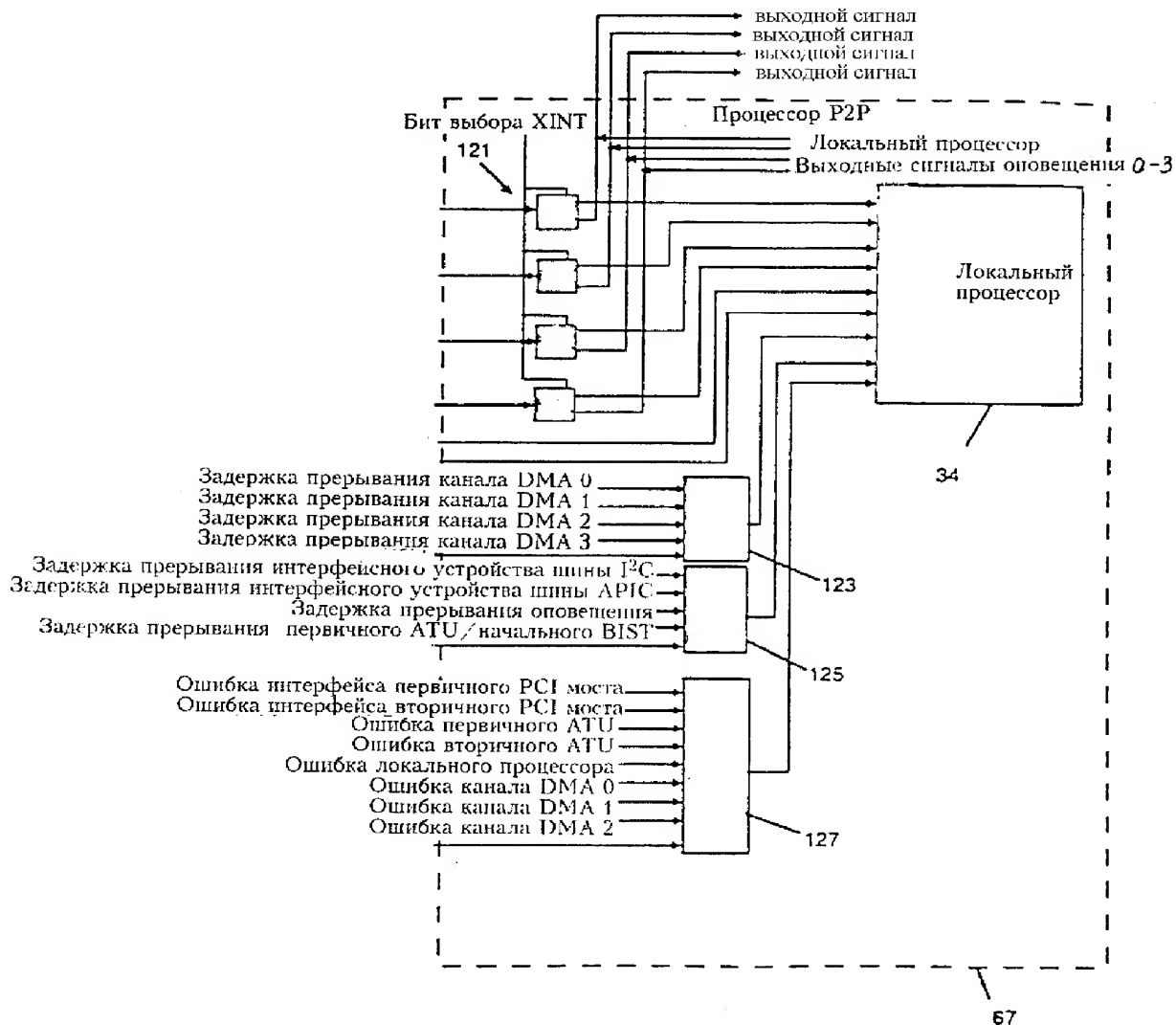




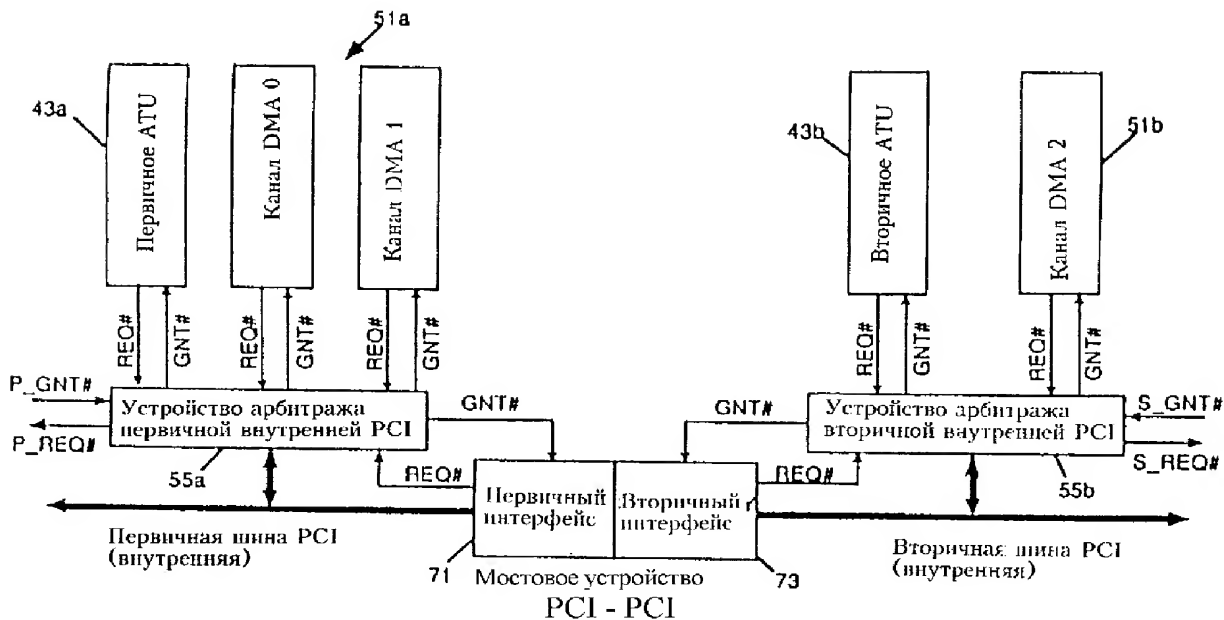
Фиг.7



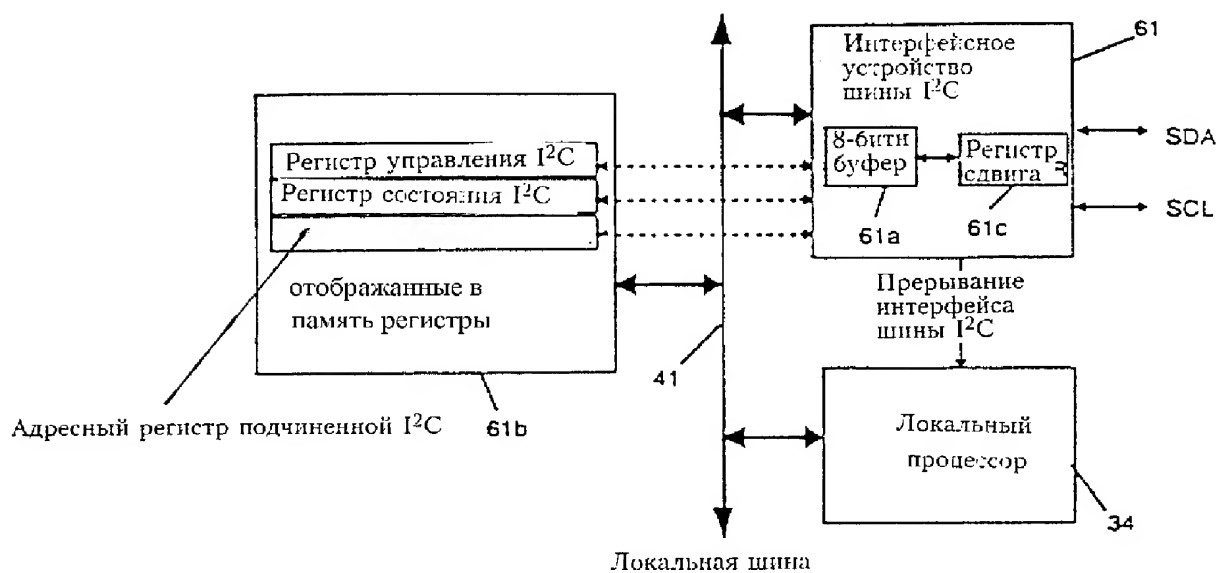
Фиг.8



Фиг.9



Фиг.10



Фиг.11